



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Aedes aegypti Mapping Tool (AaMT): Proposta de um Sistema de Geomapeamento de Focos do Aedes aegypti

Trabalho de Conclusão de Curso

Weslan Rezende Alves



São Cristóvão – Sergipe

Maio de 2020

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Weslan Rezende Alves

Aedes aegypti Mapping Tool (AaMT): Proposta de um Sistema de Geomapeamento de Focos do Aedes aegypti

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador(a): Dra. Leila Maciel de Almeida e Silva
Coorientador(a): Dra. Debora Maria Coelho Nascimento e Dra. Roseli La Corte dos Santos

São Cristóvão – Sergipe

Maio de 2020

*Este trabalho é dedicado a todas as pessoas que
sofreram de doenças epidemiológicas. Dedico, também,
a minha mãe e a meu pai que, infelizmente, não viveu
para partilhar deste meu sonho, que é o de me tornar cientista.*

Agradecimentos

Agradeço imensamente a Deus por permitir chegar ao final desta etapa da minha vida, a minha mãe, Magneide de Rezende Cunha, a pessoa que me instruiu e educou para que eu sempre lute pelos meus sonhos. Agradeço ao meu pai, Jailson Vieira Alves, por ajudar a minha mãe enquanto vivo, sempre me amando.

Quero agradecer a minha namorada, Ana Luzia Vasconcelos Barreto, que me insentivou durante toda minha trajetória do curso; sem ela esta caminhada seria mais pedregosa do que foi. Agradeço a meus irmãos, Wesley e Nany, a meu cunhado, Lucas Menezes e as minhas recém chegadas sobrinhas, ambas Maria Cecília, as quais desejo e deposito todo amor do mundo. Agradeço também a meus futuros sogro, sogra e cunhada, Maria Mirene, José Valmir e Anny Barreto, por todos momentos de alegria compartilhada.

De maneira especial quero agradecer a toda minha família, a qual compartilhei este sonho que se realiza, por todo amor a mim dedicado desde a minha infância até hoje. Agradeço a meu primo Allan Cunha, com quem dividi a alegria de ser criança, jovem e agora os conselhos compartilhados de adultos. Agradeço a minhas tias Edileuza, pelo incentivo no saber, e Sandra, pelo ensino religioso.

Agradeço também as minhas coorientadoras, professora Dra. Débora Nascimento pelo esforço e apoio na construção deste projeto, abdicando até de seu tempo particular para me orientar e também a professora Dra. Roseli Santos por seu imenso carisma e esforço em disponibilizar recursos para este projeto, tornando-o possível.

Agradeço a todos os professores do Departamento de Computação que marcaram o meu trajeto durante o curso, como o professor Bruno Prado e seu jargão "fácil demais". Agradeço a todos meus colegas de classe com quem compartilhei tarde de alegria e também agradeço a Sidney Lourdes César Souza Sa e José Oliveira Santos, da Secretaria de Saúde de Sergipe, por se disponibilizarem a ajudar na construção deste projeto.

E, em especial, quero agradecer a minha orientadora, professora Dra. Leila Silva, a qual compartilhei tardes de segunda-feira durante mais de um ano, por toda paciência em me orientar, e, mesmo quando passou por momentos difíceis, nunca deixou de auxiliar na construção deste trabalho.

*O único lugar aonde o sucesso vem
antes do trabalho é no dicionário
(Albert Einstein)*

Resumo

O âmbito desta monografia é multidisciplinar, envolvendo as áreas de Computação e Saúde. O projeto aqui desenvolvido consiste em modelar uma ferramenta de apoio ao monitoramento de focos do *Aedes aegypti*. A ferramenta proposta provê suporte automático às atividades preconizadas por duas metodologias de combate a este mosquito, a saber: Levantamento Rápido de Índice para *Aedes aegypti* e Programa Nacional de Combate da Dengue. O sistema proposto realiza a gestão dos ciclos epidemiológicos e as atividades realizadas em cada ciclo pelos municípios.

A ferramenta, quando desenvolvida, permitirá a coleta de amostras e registro dos tratamentos, realizados nos recipientes, para emissão de relatório diários e por semana epidemiológica. As amostras coletadas são examinadas e seus resultados são registrados na ferramenta para o cálculo dos Índices de Breteau e Infestação Predial. Os dados coletados durante as atividades são disponibilizados em forma de relatórios, gráficos e mapas, e podem ser exportados em formato de XLS, CSV ou JSON via plataforma FIWARE.

Para o projeto da ferramenta adotou-se a metodologia Scrum Solo e foi parcialmente implemetada usando a linguagem JavaScript com as bibliotecas React, React-Native e NodeJS.

Palavras-chave: *Aedes aegypti*, LIRAA, PNCD, Dengue, Scrum Solo, Tecnologia da Informação.

Abstract

The scope of this monograph is multidisciplinary, involving the areas of Computing and Health. The project developed here consists of modeling a tool to support the monitoring of outbreaks of *Aedes aegypti*. This tool that provides automatic support for activities recommended by two methods of fighting this mosquito: Rapid Index Survey for *Aedes aegypti* and National Program to Combat Dengue. The proposed system manages the epidemiological cycles and the activities performed in each cycle by the municipalities.

The tool, when developed, can collect samples and treatment records, executed at the recipients, for issuing daily and epidemiological week reports. The collected samples are examined and their results are recorded by the system to compute the in the Breteau index and building infestation index. The data collected during the activities are made available in the form of reports, graphs and maps and can be exported on XLS, CSV or JSON format via the FIWARE platform.

For the design of the proposed tool, the Scrum Solo methodology was adopted and the tool was partially implemented using a JavaScript language with the React, React-Native and NodeJS libraries.

Keywords: *Aedes aegypti*, LIRAA, PNCD, Dengue, Scrum Solo, Information Technology.

Lista de ilustrações

Figura 1 – Ciclo de vida da fêmea <i>Aedes aegypti</i>	20
Figura 2 – Classificação do total de municípios de acordo com o Índice Predial (IP), Brasil, 2018 e 2019	25
Figura 3 – Fluxo do processo Scrum Solo	29
Figura 4 – A arquitetura MVC	30
Figura 5 – Orion Context Broker	34
Figura 6 – Exemplo de um diagrama de casos de uso	36
Figura 7 – Exemplo de uma classe	36
Figura 8 – Exemplo de associações entre classes	38
Figura 9 – Exemplo de diagrama de atividades	39
Figura 10 – Diagrama de Caso de Uso	45
Figura 11 – Estrutura de pastas da API, aplicação WEB e <i>mobile</i>	54
Figura 12 – Visão da vistoria do Diagrama de Classes da ferramenta AaMT	55
Figura 13 – Fluxo típico da ferramenta	57
Figura 14 – Tela de login dos usuários	58
Figura 15 – Tela de cadastro de um novo ciclo	59
Figura 16 – Tela de planejamento de um estrato	60
Figura 17 – Tela de planejamento de equipes	60
Figura 18 – Tela de cadastro de município	61
Figura 19 – Tela de cadastro de localidade	62
Figura 20 – Tela de cadastro de quarteirão	63
Figura 21 – <i>Backlog</i> da <i>sprint</i> I	69
Figura 22 – <i>Backlog</i> da <i>sprint</i> II	70
Figura 23 – <i>Backlog</i> da <i>sprint</i> III	70
Figura 24 – <i>Backlog</i> da <i>sprint</i> IV	70
Figura 25 – <i>Backlog</i> da <i>sprint</i> V	71
Figura 26 – <i>Backlog</i> da <i>sprint</i> VI	71
Figura 27 – Visão de Local do Diagrama de Classe da ferramenta AaMT	83
Figura 28 – Visão de atividade do Diagrama de Classes da ferramenta AaMT	84
Figura 29 – Visão de tratamento do Diagrama de Classes da ferramenta AaMT	85
Figura 30 – Visão de amostra do Diagrama de Classes da ferramenta AaMT	86

Lista de quadros

Quadro 1 – Requisitos não funcionais da ferramenta AaMT	42
Quadro 2 – Requisitos funcionais da ferramenta AaMT	43
Quadro 3 – <i>Product Backlog</i>	52
Quadro 4 – <i>Backlog da sprint I</i>	53
Quadro 5 – <i>Backlog da sprint II</i>	54

Lista de códigos

Código 1 – Código JavaScript - Consultando entidade por conteúdo dos atributos . . .	87
Código 2 – Código JSON - Resultado por consulta de conteúdo dos atributos	88

Lista de abreviaturas e siglas

AaMT	Aedes aegypti Mapping Tool
ACE	Agente de Controle a Endemias
API	<i>Application Programming Interface</i>
DMO	Departamento de Morfologia
EAP	Estrutura Analítica do Projeto
GE	<i>Generic Enablers</i>
IB	Índice de Breteau
IoT	<i>Internet of Things</i>
IP	Índice Predial
ITR	Índice por Tipo de Recipiente
LEPaT	Laboratório de Entomologia e Parasitologia Tropical
LIRAA	Levantamento Rápido de Índice para Aedes aegypti
MVC	Model-View-Controller
OO	Orientação a Objeto
PNCD	Programa Nacional de Controle da Dengue
RF	Requisitos funcionais
RNF	Requisitos não funcionais
SE 12	Doze primeiras semanas do ano
SUS	Sistema Único de Saúde
TIC	Tecnologia da Informação e Comunicação
UFS	Universidade Federal de Sergipe
UML	Unified Modeling Language

Sumário

1	Introdução	14
1.1	Objetivo Geral	16
1.2	Objetivos específicos	16
1.3	Metodologia	17
1.4	Organização	17
2	Fundamentação Teórica	19
2.1	Conceitos Básicos na Área de Saúde	19
2.1.1	Ciclo de vida do <i>Aedes aegypti</i>	19
2.1.2	A metodologia LIRAA	21
2.1.2.1	Índices	21
2.1.2.2	Tipos de criadouros	22
2.1.2.3	Planejamento e operação da LIRAA	23
2.1.3	O PNCD	24
2.1.3.1	Operação do PNCD	26
2.2	Conceitos básicos na área da Computação	26
2.2.1	Metodologia Scrum Solo	26
2.2.2	Arquitetura de <i>Software</i> MVC	29
2.2.3	Linguagens e Ferramentas de Desenvolvimento	30
2.2.3.1	React	31
2.2.3.1.1	React Native	31
2.2.3.2	Node.js	31
2.2.3.3	A plataforma FIWARE	32
2.2.3.4	<i>Unified Modeling Language</i> - UML	34
2.2.3.4.1	Diagrama de Caso de Uso	35
2.2.3.4.2	Diagrama de Classes	36
2.2.3.4.3	Diagrama de atividades	38
2.3	Trabalhos Relacionados	39
3	Proposta da Ferramenta AaMT	41
3.1	Levantamento de Requisitos	42
3.1.1	Definir Ciclo	46
3.1.2	Manter Atividade	47
3.1.3	Planejar Atividade	47
3.1.4	Definir Trabalho Diário	48
3.1.5	Iniciar Trabalho Diário	49

3.1.6	Finalizar Trabalho Diário	49
3.1.7	Realizar Vistoria	50
3.1.8	Realizar Exame	51
3.2	<i>Backlog</i> do Produto e <i>Sprints</i>	51
3.3	Projeto da Ferramenta	54
3.3.1	Diagrama de Classes	55
3.3.2	Diagrama de Atividades	56
3.4	Principais funcionalidades	58
4	Conclusão	64
	Referências	66

Apêndices 68

APÊNDICE A	Documentação complementar da ferramenta AaMT	69
A.1	<i>Sprints</i>	69
A.2	Casos de Uso	71
A.2.1	Manter Município	71
A.2.2	Manter Usuário	72
A.2.3	Manter Localidade	72
A.2.4	Manter Zona	73
A.2.5	Manter Laboratório	73
A.2.6	Manter Quarteirão	73
A.2.7	Manter Rua	74
A.2.8	Manter Imóvel	74
A.2.9	Visualizar Resultado da Atividade	75
A.2.10	Visualizar Resultado do Ciclo	75
A.2.11	Manter Regional de Saúde	75
A.2.12	Relatório por Ciclo	76
A.2.13	Manter Atividade para Município	76
A.2.14	Definir Perfis	77
A.2.15	Visualizar resumo do trabalho diário do município	77
A.2.16	Visualizar Resumo do trabalho diário do município	77
A.2.17	Visualizar Resultado da Atividade no Município	78
A.2.18	Acompanhar demandas por Laboratório	78
A.2.19	Resumo do Trabalho Diário da Equipe	78

A.2.20	Resumo do Trabalho Diário	79
A.2.21	Registrar Dados Abertos Na FIWARE	79
A.2.22	Imprimir formulário de vistoria	79
A.2.23	Visualizar Boletim de Campo	80
A.2.24	Visualizar Relatório Semanal	80
A.2.25	Enviar Amostra para Laboratório	81
A.2.26	Registro do trabalho diário para o agente	81
A.2.27	Visualizar Resumo do Trabalho diário da Equipe	81
A.2.28	Visualizar resumo do resultado de laboratório de suas amostras	82
A.2.29	Acompanhar demandas por laboratório pelo supervisor	82
A.3	Diagrama de Classes	82
APÊNDICE B	Exemplo de código, em jQuery, de requisições a API ORION . . .	87
Anexos		89
ANEXO A	Glossário do Boletim de Campo e Laboratório do Levantamento Rápido de Índices - LIRAa	90
ANEXO B	Glossário do Boletim de Campo do Programa Nacional de Controle da Dengue - PNCD. Serviço Antiveterial	93

1

Introdução

O Brasil possui o Sistema Único de Saúde (SUS), o qual é responsável por garantir o atendimento gratuito a todos os cidadãos do país. O SUS foi criado em 1988, juntamente com o artigo 196 da constituição que diz “A saúde é direito de todos e dever do Estado”. Foi um marco na vida dos cidadãos, pois o atendimento gratuito nem sempre era possível. Sendo assim, para proporcionar tal benefício à população o SUS deve atender a toda população com seus serviços ou pela contratação de convênios ou terceirizados.

Apesar de seu surgimento ter sido um fator de grande valia para a população, ainda existem aspectos a serem trabalhados neste sistema para torná-lo mais eficaz. Apesar do SUS ser modelo de referência de saúde pública em todo o mundo, sua implantação no Brasil precisa evoluir. Em 2011, em uma escala de 0 a 10, somente dois estados atingiram uma avaliação média entre 6 a 6,99, três atingiram avaliação entre 4 a 4,99 e os demais estados obtiveram média entre 5 a 5,99, indicando, assim, que ainda há muito o que melhorar ([SAGE, 2011](#)).

Um dos encargos atribuídos ao SUS é o controle de doenças, como o combate a doenças transmitidas por hematófagos¹. Uma das mais preocupantes doenças transmitidas por hematófagos é a Dengue, doença transmitida pelo mosquito *Aedes aegypti*. Além da Dengue, este mosquito também é capaz de transmitir doenças como *Zika*, *Chikungunya* e Febre Amarela, o que torna o combate a este vetor ainda mais relevante. Para controlar a endemia destas doenças é necessário combater diretamente o mosquito transmissor, em que o papel de controle vetorial é atribuído aos Agentes de Controle a Endemias (ACEs).

O boletim epidemiológico lançado pelo Ministério da Saúde ([MS, 2019a](#)), que faz comparação das 12 primeiras semanas (SE 12) de 2019 (30/12/2018 à 23/03/2019) com o mesmo período de 2018, relata que houve um aumento de 282% no número de casos prováveis² de Dengue em relação a 2018, um significativo acréscimo de casos. No entanto, apesar desta

¹ Hematófagos: a forma como denominamos os animais que se alimentam de sangue de outros animais.

² Casos prováveis: entende-se por casos prováveis todos os casos notificados, excluindo-se os descartados.

informação preocupante, o Ministério da Saúde informa: “ressalta-se que mesmo com aumento no número de casos, a taxa de incidência de 2019 está dentro do canal endêmico, ocorrência esperada para o período, obtido a partir da série histórica dos últimos 8 anos...”. Sendo assim, apesar do notável aumento do casos de doenças transmitidas pelo mosquito *Aedes aegypti*, segundo o Ministério da Saúde, o Brasil ainda não se encontra em situação de epidemia.

De acordo com o boletim (MS, 2019a), em 2019 foram registrados 273.193, casos prováveis de Dengue no país, enquanto no mesmo período de 2018 foram registrados 71.525 casos prováveis. Se somarmos os casos prováveis de Dengue, *Chikungunya* e *Zika* em 2019 foram registrados 290.899 casos na SE 12, enquanto que em 2018, no mesmo período, apenas 100.858.

Para combater o mosquito *Aedes aegypti*, e consequentemente reduzir os números de casos prováveis, os ACEs seguem duas metodologias que são: Levantamento Rápido de Índices para *Aedes aegypti* (LIRAA) e o Programa Nacional de Controle da Dengue (PNCD). Ambas trabalham de forma semelhante, pois ambas necessitam que os ACEs vão a campo preencher formulários nas casas dos cidadãos. No entanto, os objetivos das metodologias diferem. O LIRAA tem como finalidade realizar índices, por amostragem, que condiz com a realidade da localidade com uma margem de erro aceitável (LIRAA, 2013) e sua execução, normalmente, não dura mais que uma semana. Já o PNCD executa atividade de combate diário ao mosquito *Aedes aegypti* (PNCD, 2002), em que os ACEs vão a campo, com larvicidas para usar em recipientes, a fim de evitar futuros focos, ou realizar a eliminação do mesmo.

Embora o uso dessas metodologias seja amplamente disseminado no Brasil, é realizado fazendo uso de papeis para preenchimento de formulários, o que gera problemas como: espaço para armazenamento, erro humano de preenchimento, desperdício de tempo para as pessoas que precisam contabilizar e/ou transcrever as informações para os sistemas requeridos.

Com o advento da tecnologia, é possível agilizar o processo, fazendo com que o tempo dos ACEs seja melhor aproveitado, permitindo assim, que a equipe passe mais tempo combatendo o *Aedes aegypti* e não realizando medidas administrativas, como redigir formulários. Nesta direção, o desenvolvimento de aplicações que façam uso de aparelhos celulares tornou-se uma ferramenta importante, pois desta forma, é possível suprimir o uso do papel durante as vistorias, melhorando e sanando alguns dos problemas aqui anteriormente elencados.

Atualmente já existem algumas aplicações que auxiliam os Agentes de Controle a Endemias (ACE) no controle vetorial dos mosquitos como o *Aedes aegypti*. Um exemplo destes aplicativos é o Dengue-MI (EIRAS; RESENDE, 2009) que, em tempo real, realiza o mapeamento das áreas, fazendo uso de uma palheta de cores, de acordo com o resultados obtidos nas amostras através do uso de armadilhas que capturam fêmeas grávidas do mosquito *Aedes aegypti*.

Aplicações como a Dengue-MI coletam dados de extrema importância e é interessante que estes dados possam ser transmitidos o mais rápido possível, tornando o desenvolvimento de tais ferramentas complexas. Soluções como a *Application Programming Interface* (API -

Interface de programação de aplicativos) ORION da FIWARE podem auxiliar neste quesito. Esta API possui a capacidade de receber grandes volumes de dados de contexto e disponibilizá-los rapidamente para outras plataformas, como a CKAN, uma plataforma para criação de portal de dados abertos. A FIWARE (FIWARE, 2019) é uma plataforma de código aberto e provê serviços como *Internet of Things* (IoT - Internet das Coisas), suporte a transferência de grande volume de dados, gerenciamento de informações de contexto, segurança e ambiente de criação de aplicativos para profissionais sem experiência em desenvolvimento.

O contexto deste trabalho foi agregar tecnologia da informação ao combate e estudo epidemiológico de doenças transmitidas pelo mosquito *Aedes aegypti*. Nesta direção foi proposta a ferramenta *Aedes aegypti* Mapping Tool (AaMT) para dar suporte automático às duas metodologias de combate ao mosquito, LIRAA e PNCD. A gestão do processo do combate ao *Aedes aegypti* realizado no municípios foi projetada no planejamento da ferramenta, de forma que as atividades realizadas pelos ACEs possam ser acompanhadas de perto pelos seus gestores, ao mesmo tempo em que dados abertos possam ser disponibilizados para estudo epidemiológico do mosquito, para estudos conduzido pelo Laboratório de Entomologia e Parasitologia Tropical (LEPaT) do Departamento de Morfologia (DMO) da Universidade Federal de Sergipe.

O LEPaT, chefiado pela Dra. Roseli La Corte dos Santos, realiza pesquisas de campo nos municípios para coleta de dados específicos, que não são coletados pelas metodologias aplicadas pelos ACEs. Como, por exemplo, a quantidade de mosquitos *Aedes aegypti* adultos encontrados no município. Buscando auxiliar na solução deste problema, a ferramenta propõe disponibiliza em formato de dados abertos informações de vistoria, usando a plataforma FIWARE. Sendo assim, acessível para qualquer grupo de pesquisa ou órgão governamental que deseje utilizar a informação coletada.

1.1 Objetivo Geral

Projetar e prototipar um sistema para registro, mapeamento e divulgação dos focos do mosquito *Aedes aegypti*, visando auxiliar a tomada de decisão dos ACEs no combate vetorial do mosquito, bem como prover suporte automático às atividades relacionadas ao estudo epidemiológico das doenças transmitidas por este vetor e disponibilizar os dados coletados na forma de dados abertos.

1.2 Objetivos específicos

- Estudar metodologias usadas no combate ao mosquito, como Levantamento Rápido de Índice para *Aedes aegypti* e o Programa Nacional de Controle da Dengue, para compreensão do problema;
- Especificar, projetar e prototipar a ferramenta AaMT, aqui proposta;

- Propor a disponibilizar dados, coletados pelos ACEs, na forma de dados abertos, usando a plataforma FIWARE.

1.3 Metodologia

Este trabalho compreendeu as seguintes atividades principais:

1. Revisão bibliográfica sobre *Aedes aegypti* e metodologias de monitoramento do mosquito;
2. Revisão bibliográfica do conteúdo técnico em computação necessário para o desenvolvimento deste trabalho;
3. Modelagem e levantamento de requisitos do sistema;
4. Visita de campo para a percepção da atuação dos ACEs no combate ao mosquito;
5. Projeto e implementação do protótipo da aplicação segundo os preceitos da metodologia Scrum Solo.

As atividades 1 e 2 foram realizadas através dos estudos das metodologias LIRAA, PNCD e conhecimento do mosquito *Aedes aegypti*, descritas na [seção 2.1](#). Baseados no estudos acerca do problema e no conhecimento tecnológico do autor foi identificado a necessidade do uso de algumas tecnologias recentes como React, NodeJS e React-Native para o desenvolvimento do sistema, bem como a plataforma FIWARE para a disponibilização dos dados na forma de dados abertos.

Para a realização da atividade 3, foi estabelecida a relação entre as metodologias LIRAA, PNCD, com as atividades dos ACEs e as atividades da professora Dra. Roseli Lacorte dos Santos no LEPaT. Foram elicitados os requisitos do sistema capazes de atender as demandas dos ACEs e do grupo de pesquisa do LEPaT. Os requisitos foram validados pela profa. Dra. Roseli La Corte dos Santos e por membros da Secretaria de Estado da Saúde de Sergipe.

A atividade 4 foi realizada no município de Simão Dias. Durante esta atividade foi possível observar como são realizadas as atividades dos ACEs dentro do Centro de Combate a Endemias e em campo, realizando vistorias nos imóveis do município de Simão Dias.

A atividade 5 foi desenvolvida seguindo os preceitos da metodologia Scrum Solo, descrita em [seção 2.2.1](#), e publicado no *link* [<https://aamt.herokuapp.com/>](https://aamt.herokuapp.com/).

1.4 Organização

Este trabalho é composto por mais três capítulos onde são descritas a fundamentação teórica, a ferramenta AaMT e a conclusão.

O [Capítulo 2](#) compreende principais conceitos das duas áreas relacionadas ao projeto, **Saúde** e **Computação**, já que o trabalho é multidisciplinar. Na área de Saúde, o ciclo de desenvolvimento do *Aedes aegypti* e as metodologias utilizadas pelos ACEs, LIRAA e PNCD, são descritos. Na área de Computação, é descrita a metodologia de desenvolvimento de sistema, Scrum Solo, a plataforma FIWARE, ferramentas e linguagens utilizadas no projeto. Neste capítulo também são apresentados alguns trabalhos relacionados ao tema desta monografia.

O [Capítulo 3](#) apresenta a descrição da ferramenta AaMT e, por fim, no [Capítulo 4](#) são apresentadas algumas considerações finais e trabalhos futuros.

2

Fundamentação Teórica

Como mencionado no [Capítulo 1](#), este trabalho é multidisciplinar pois trata-se do desenvolvimento de uma ferramenta para a área de saúde pública. Assim, a seguir são apresentados os conceitos básicos tanto na área de Saúde ([seção 2.1](#)), como na área de Computação ([seção 2.2](#)), necessários para o entendimento desta monografia.

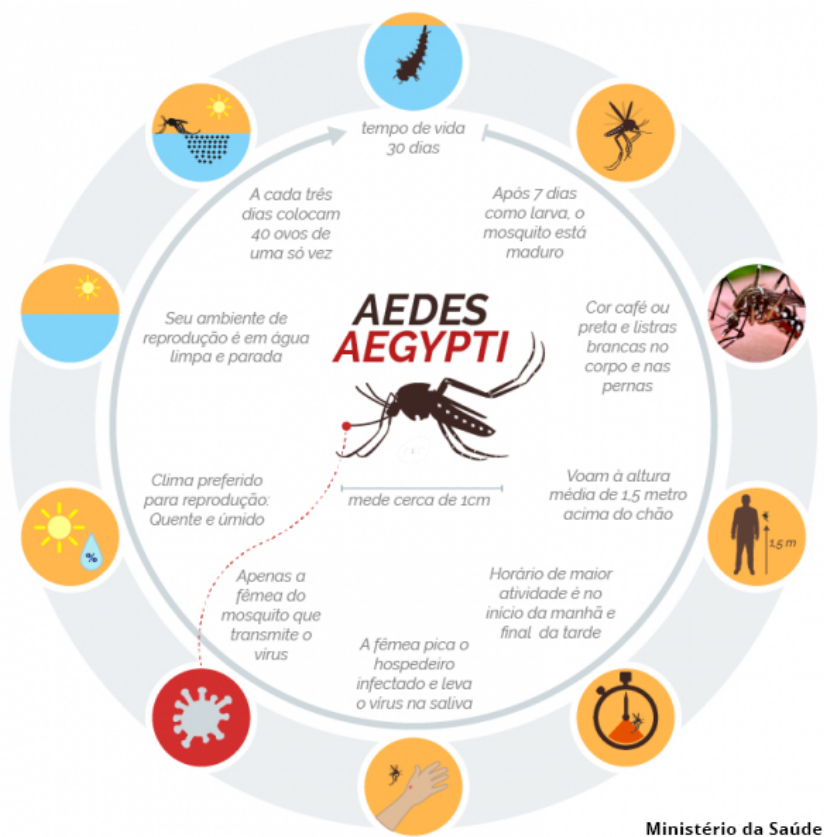
2.1 Conceitos Básicos na Área de Saúde

Nesta seção são descritos brevemente o ciclo do *Aedes aegypti* ([seção 2.1.1](#)), mosquito alvo deste trabalho, bem como as metodologias empregadas pelo SUS para o combate a este mosquito ([seções 2.1.2 e 2.1.3](#)).

2.1.1 Ciclo de vida do *Aedes aegypti*

O mosquito *Aedes aegypti*, transmissor das doenças como Dengue, *Chikungunya*, *Zika* e Febre Amarela, é um vetor que assola as regiões brasileiras há muito anos e ainda continua a assombrar a população com doenças capazes de levar a óbito caso não tratadas corretamente. Portanto, faz-se necessário entender como o mosquito vive e se reproduz.

O ciclo de vida do mosquito *Aedes aegypti* passa por quatro fases: ovo, larva, pupa e adulto. A fêmea deposita seus ovos em ambientes úmidos e quentes próximo da linha d'água dos recipientes como pneu, garrafas, latas, etc. Normalmente os ovos são postos em águas limpas e paradas, mas estudos apontam a existência de focos do mosquito em água suja ([MS, 2019b](#)).

Figura 1 – Ciclo de vida da fêmea *Aedes aegypti*

Fonte: (MS, 2019b)

A figura 1 apresenta o ciclo de vida da fêmea do *Aedes aegypti*. Após o depósito dos ovos, estes aguardam a presença de água para eclodirem. Depois da eclosão, o *Aedes aegypti* encontra-se na fase de larva, onde passa cerca de cinco dias até se tornar pulpa, e então, passar mais dois ou três dias até chegar à fase adulta. É durante a fase adulta que o mosquito está maduro e pronto para transmitir doenças. O maior perigo do mosquito são as fêmeas, pois são elas as transmissoras das arboviroses¹, já que a sua alimentação é o sangue humano e, caso estejam infectadas, é através da saliva que nos contaminam durante a picada. A cada três dias elas conseguem colocar cerca de quarenta ovos, reiniciando o ciclo.

Para o controle deste vetor são necessárias medidas preventivas que devem ser adotadas em residências. O Ministério da Saúde lista dezenove ações que a população deve adotar para combater o mosquito (MS, 2019b), algumas delas são:

- Manter bem tampado tonéis, caixas e barris de água;
- Encher pratinhos de vasos com areia até a borda ou lavá-los uma vez por semana;

¹ Arboviroses: são as doenças causadas pelos chamados arbovírus, que incluem o vírus da Dengue, Zika vírus, febre *Chikungunya* e febre amarela. A classificação "arbovírus" engloba todos aqueles transmitidos por artrópodes, ou seja, insetos e aracnídeos (como aranhas e carrapatos).

- Acondicionar pneus em locais cobertos.

Ações como essas servem para eliminar focos das proximidades e devem ser adotadas por todos, pois o mosquito *Aedes aegypti* pode infectar pessoas em um raio de 100 metros de onde ocorreu a desova.

Além do combate realizado pela população, existem programas de combate ao mosquito, realizado pelos ACEs, como Levantamento Rápido de Índices para *Aedes Aegypti* e o Programa Nacional de Controle da Dengue. Estas atividades possuem regras e normas adotadas, diariamente ou em ciclos, pelos ACEs e servem para maximizar os efeitos do combate as arboviroses.

Nas próximas seções as metodologias LIRAA e PNCD são detalhadas.

2.1.2 A metodologia LIRAA

A LIRAA é a metodologia adotada pelos serviços de saúde baseada em amostragens para realização de vistorias em campo sobre a existência do mosquito nas fases de vida citadas, visando facilitar a obtenção de informações através de métodos simplificados que permitam estimativas com erros aceitáveis.

De forma diferente da metodologia PNCD, esta metodologia disponibiliza a capacidade de geração de dados amostrais em um curto intervalo de tempo, evidenciando a situação atual da localidade. Isto maximiza o uso dos recursos humanos e auxilia a gestão dos gastos em locais apontados como críticos pela LIRAA.

Apesar de ser um sistema rápido, sua implantação deve adotar normas para que os índices amostrais cheguem o mais perto possível da situação real da localidade. Alguns dos índices que são possíveis levantar com a metodologia LIRAA são o Índice de Breteau, o Índice Predial e o Índice por Tipo de Recipiente (LIRAA, 2013).

A seguir são detalhados os índices, os tipos de criadouros considerados pela metodologia, bem como o planejamento e operação da LIRAA.

2.1.2.1 Índices

Embora os índices a seguir sejam de grande importância para os agentes fica a ressalva que o Índice Predial não leva em conta o número de recipientes positivos e nem o seu potencial de produtividade, assim como o de Breteau também não considera o potencial de produtividade dos criadouros. Estes índices são calculados da seguinte forma:

- **Índice Predial (IP):** método para calcular o percentual de imóveis positivos. O imóvel é considerado positivo se pelo menos um dos recipientes vistoriados no imóvel der positivo para a presença de larva do mosquito. A LIRAA segue os seguintes critérios de avaliação

deste índice: menor que 1 - satisfatório, entre 1 a 3,9 - alerta e acima de 3,9 - risco. O índice é dado por:

$$IP = \text{Imóveis Positivos} / \text{Imóveis Pesquisados} * 100$$

- **Índice de Breteau (IB):** expressa a relação de recipientes positivos em relação aos imóveis pesquisados, expresso por 100 imóveis:

$$IB = \text{Recipientes Positivos} / \text{Imóveis Pesquisados} * 100$$

- **Índice por tipo de recipiente (ITR):** percentual de um tipo de recipiente em relação ao número de recipientes positivos totais:

$$X = \text{Número de recipientes positivos por tipo escolhido}$$

$$ITR = X / \text{Total de recipientes positivos} * 100$$

2.1.2.2 Tipos de criadouros

A metodologia LIRAA considera os seguintes tipos de criadouros:

- **Grupo A:** armazenamento de água;
 - **A1:** depósito de água elevado, cisternas ou mina d'água, caixas d'águas, tambores, depósitos de alvenaria;
 - **A2:** depósitos em nível do solo, tonel, tambor, barril, tina, depósitos de barro, cisternas, caixas d'águas, captação de água em poço, cacimba, cisterna;
- **Grupo B:** depósitos móveis, vasos, frascos com água, pratos, garrafas retornáveis, pingadeira, recipientes de degelo em geladeiras, bebedouros, pequenas fontes ornamentais, materiais em depósitos de construção, objetos religiosos;
- **Grupo C:** depósitos fixos, tanques em construção civil, borracharias e hortas, calhas, lajes, toldos em desnível, ralos, sanitários em desuso, piscinas não tratadas, fontes ornamentais, cacos de vidro em muros, outras obras e adornos arquitetônicas;
- **Grupo D:** passíveis de remoção;
 - **D1:** pneus e materiais rodantes;
 - **D2:** resíduos sólidos (recipientes plásticos, garrafas PET, latas), sucatas e entulhos de construção;
- **Grupo E:** naturais, axilas de folhas, buracos em árvores e rochas, restos de animais (cascas, carapaças, etc.).

2.1.2.3 Planejamento e operação da LIRAA

Antes de iniciar o levantamento de índices é necessário pelo menos duas semanas de planejamento da atividade (LIRAA, 2013). Convém ressaltar que em municípios em estado de epidemia não pode ser feito a LIRAA, pois todos os esforços dos ACEs devem estar voltados para o combate à epidemia. Outra regra preconizada pela metodologia é que todas as atividades realizadas pelos agentes devem ser suspensas, pois quaisquer ações feitas em paralelo a LIRAA podem influenciar nos índices levantados pela LIRAA, podendo afetar a realidade da localidade estudada.

Para execução do método é necessário que a equipe de ACEs saiba seu papel no levantamento, portanto existem algumas funções para executar as tarefas da LIRAA:

- **Coordenador:** responsável por procurar apoio para o levantamento do método, por levantar informações necessárias para a operação, por definir cargos dos membros como supervisores, agentes, laboratoristas, etc;
- **Supervisor:** responsável por definir equipes e distribuí-las nos campos de trabalho, por abastecer ativos dos ACEs como: imprimir formulários de vistoria, folhetos de campanhas, preparar larvicida, além de supervisionar atividades, encaminhar amostras para os laboratórios, etc;
- **ACE:** responsável por visitar entre 20 à 25 imóveis por dia, preenchendo formulário de vistoria em cada imóvel, coletando amostras, etc.

Após a definição dos responsáveis por cada papel começa o trabalho de campo. Fazendo uso do sistema informatizado da LIRAA o coordenador insere no software todos os bairros do município, o qual retorna de forma aleatória os bairros sorteados para realização da LIRAA (LIRAA, 2013).

O sistema informatizado do LIRAA é utilizado pelos coordenadores para o cálculo de quais bairros ele deve trabalhar. Nele também é registrado todos os registros de tempos e resultados das amostras centralizando as informações coletadas pelos municípios.

Com os bairros em mãos e distribuídos entre os supervisores, as equipes de cada supervisor são definidas e vão a campo para realizar as vistorias nos imóveis existentes nos bairros definidos preenchendo o formulário exibido no [Anexo A](#).

Todo processo de preenchimento do formulário é feito pelos ACEs de forma manual e com uso de papel. Ao final de cada dia, a LIRAA pede que os supervisores analisem todos os boletins de campo para verificação de erros ou dados incompletos, para correção. Além disto, outras atribuições dos supervisores são: receber as amostras coletadas em campo, enviar as amostras aos laboratórios para análise. Os supervisores realizam o somatório manual de todas as

informações contidas nos boletins, juntamente com os resultados já realizados de amostras, e fazem a inserção no sistema da LIRAA.

Ao final de toda a coleta de dados, alguns municípios contabilizam o número de imóveis, que pode ser retirado dos boletins contando-se as linhas preenchidas pelos agentes, pois cada linha representa um imóvel pesquisado. Outro número importante para o cálculo dos índices é o número de imóveis positivos, o qual depende dos resultados dos laboratórios, pois a partir deste dado contabilizam-se as amostras que deram soro tipo positivo. Estes dados subsidiam os cálculos dos índices, mencionados na [seção 2.1.2.1](#), para obter os indicadores do município.

O fato de todo esse processo ser realizado de forma manual, acarreta em sérios problemas, como possíveis erros humanos, tempo de espera, etc. A aplicação aqui proposta a **AaMT**, pode mitigar parte destes problemas, se vier a ser usada nos municípios para automatizar os processos realizados pela LIRAA.

2.1.3 O PNCD

Criado em 2002, o PNCD assemelha-se à LIRAA e foi implantado para enfatizar o combate ao vetor da Dengue. Este programa baseia-se em oito aspectos ([PNCD, 2002](#)):

1. Elaboração de programas permanentes;
2. Desenvolvimento de campanhas de informação e de mobilização das pessoas;
3. Fortalecimento da vigilância epidemiológica e entomológica para ampliar a capacidade de predição e detecção precoce de surto de doenças;
4. Melhoria da qualidade do trabalho de campo de combate ao vetor;
5. Integração das ações de combate da Dengue na atenção básica;
6. Utilização de instrumentos legais que facilitem o trabalho do poder público na eliminação de criadouros em imóveis comerciais, casas abandonadas, etc.;
7. Atuação multissetorial por meio do fomento à destinação adequada de resíduos sólidos e a utilização de recipientes seguros para armazenagem de água;
8. Desenvolvimento de instrumentos mais eficazes de acompanhamento e supervisão das ações desenvolvidas pelo Ministério da Saúde, estados e municípios.

Para a implantação destes aspectos, o PNCD elaborou uma escala de prioridade, levando-se em conta a região e a população onde o programa vai ser aplicado, da seguinte forma:

1. Capital de estado e sua região metropolitana;

2. Município com população igual ou superior a 50.000 habitantes;
3. Municípios receptivos à introdução de novos sorotipos de Dengue (fronteiras, portuários, núcleos de turismo, etc.).

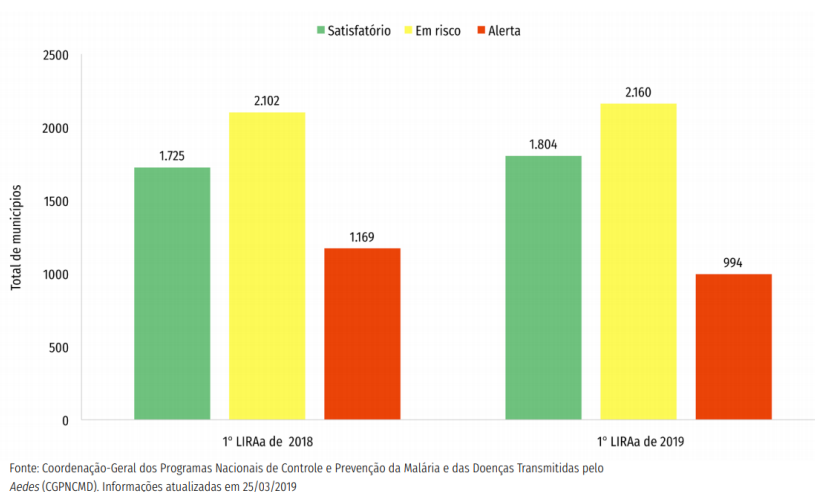
Os objetivos do programa com esses aspectos e priorização são para a redução da infestação do vetor da Dengue, redução da incidência da Dengue e diminuição da letalidade da doença. A criação do programa também gerou metas a serem alcançadas, a longo prazo, durante as atividades realizadas são:

- Reduzir a menos de 1% a infestação predial em todos os municípios;
- Reduzir em 50% o número de casos de 2003 em relação a 2002 e, no anos seguintes, 25% a cada ano;
- Reduzir a letalidade por febre hemorrágica de Dengue a menos de 1%.

Atualmente as atividades do PNCD estão amplamente difundidas entre os estados. Em particular, no município de Simão Dias, local onde foi feito os estudos iniciais da criação da AaMT, os agentes realizam o programa diariamente.

Apesar do esforço para realização do programa em todo o Brasil, o boletim de 2019 do Ministério da Saúde (MS, 2019a), a partir de dados gerados pela metodologia LIRAA aplicada em 2018 e 2019, apresentou a situação exposta na Figura 3, onde pode-se observar a classificação dos municípios de acordo com o índice predial.

Figura 2 – Classificação do total de municípios de acordo com o Índice Predial (IP), Brasil, 2018 e 2019



Fonte: Coordenação-Geral dos Programas Nacionais de Controle e Prevenção da Malária e das Doenças Transmitidas pelo Aedes. Informações atualizadas em 25/03/2019

Na [Figura 2](#), em 2019 observa-se que de um total de 4958 município existem 3154 que ainda não possuem IP menor que 1%. Portanto, aproximadamente 63,6% dos municípios brasileiros ainda não atingiram a meta do PNCD, que é obter 1% de IP em todos os municípios do país.

2.1.3.1 Operação do PNCD

Assim como na metodologia LIRAA, para aplicar o PNCD os ACEs também vão a campo preenchendo os dados do formulário do [Anexo B](#), porém os objetivos das duas metodologias são distintos. Enquanto o LIRAA visa levantar dados importantes da região de pesquisa, o PNCD faz um combate diário contra o mosquito, pois os agentes vão a campo com larvicida para despejar em recipientes que podem vir a ter foco do mosquito. Esta ação serve para combater criadouro em água limpa, e caso seja encontrado foco no recipiente, são coletadas amostras, despejada a água e é feita higienização do recipiente.

Apesar destas diferenças os processo são manuais em ambas metodologias, requerendo muito esforço para preenchimento e levantamento de dados do município.

Assim como o LIRAA, há necessidade de automatizar as tarefas dos ACEs no PNCD, para gerar melhor aproveitamento de tempo dos agentes e eliminar a necessidade de uso do papel, que gera problemas de armazenamento e coleta das informações por parte dos supervisores.

2.2 Conceitos básicos na área da Computação

Atualmente, dados já estão sendo usados como produto por grandes empresas para geração de capital. Com o atual crescimento da IA, que necessita de grande volumes de dados para aprendizado, bases que contenham dados relevantes são de grande interesse, inclusive para as pesquisas científicas.

O mesmo podemos dizer para os dados coletados por ACEs, pois estes podem fundamentar não só ações de gestão mas também pesquisas científicas na área de saúde pública. Assim, o uso de Tecnologia da Informação e Comunicação (TIC) na coleta e análise destes dados é de suma importância. Nesta seção são abordados os elementos necessários para o entendimento da parte computacional deste projeto.

2.2.1 Metodologia Scrum Solo

Atualmente a Metodologia Ágil Scrum tornou-se referência no mercado de trabalho, pois é frequentemente usada em projetos de desenvolvimento de produtos de software. Segundo ([SRIVASTAVA; BHARDWAJ; SARASWAT, 2017](#)) “cerca de 82% dos entrevistados, entre

5000 participantes, usam Scrum e 11% estão começando a usar o Scrum”, isto significa que as empresas buscam aplicar ou estão começando a usar o Scrum no desenvolvimento de seus *softwares*. Isto acontece devido a sua capacidade de se adaptar aos modelos de negócios, além de ser flexível a mudanças durante a sua execução.

O Scrum é aplicado para equipes, no entanto, existem projetos onde a mão de obra se limita a uma pessoa como: projetos pessoais, Trabalhos de Conclusão de Curso e autônomos. Com o objetivo de atender este público, o Scrum Solo (PAGOTTO et al., 2016) uniu boas práticas do *framework* Scrum (SRIVASTAVA; BHARDWAJ; SARASWAT, 2017) e do *Personal Software Process* (PSP) de forma a tentar garantir qualidade e agilidade no desenvolvimento de software com um desenvolvedor apenas.

O Scrum Solo relaciona atividades, atores e artefatos² em processos bem definidos para entrega parcial, a cada *sprint*, e total, ao final do projeto. A seguir, os elementos do Scrum Solo são brevemente descritos.

- **Atividades:**

- **Requirement:** nesta atividade é definido o cliente, o escopo do projeto e o *product backlog*. Os artefatos gerados durante a execução desta atividade são: *product backlog* e o protótipo do produto;
- **Sprint:** objetiva executar um conjunto de demandas do *product backlog* com uma duração máxima de uma semana. Ao final de uma *sprint*, pode ser realizado uma *oriented meeting* (reunião de orientação) entre o desenvolvedor e o orientador para orientações, caso seja necessário. Os artefatos gerados ao final da *sprint* são: *sprint backlog*, produto, ata da reunião de planejamento e planta de desenvolvimento;
- **Deployment:** Nesta etapa é usada a planta de desenvolvimento para disponibilização do produto para o cliente. Os artefatos gerados são: produto e ata da reunião de validação;
- **Management:** objetiva planejar, monitorar e controlar o desenvolvimento do produto. Os artefatos gerados ao fim desta atividade são: estrutura analítica do projeto (EAP), cronograma, planilha de custo e planilha de controle.

- **Atores:**

- **Product Owner:** proprietário do produto;
- **Desenvolvedor individual:** responsável por executar o processo e contruir o produto;
- **Orientador:** pessoa que entende a fundo os processos do *software* e as tecnologias aplicadas para a construção do produto;

² Artefato: segundo o Scrum Solo artefatos são documentos gerados apartir da execução de uma atividade usados para comprovar execução da atividade ou dados para melhorias futuras na aplicação da metodologias visando agilizar o processo.

- **Grupo de Validação:** usuários responsáveis por testar e validar o produto;
- **Artefatos:**
 - **Escopo:** descreve os principais características do *software* de acordo com o mapeamento do problema do *product owner*;
 - **Protótipo de Software:** coleção dos modelos de telas do *software* e seus relatórios;
 - **Product Backlog:** lista de funcionalidade que devem ser desenvolvidas no *software*, essa lista deve se limitar a funcionalidade provida do código do *software*;
 - **Repositório do processo:** local em nuvem, onde os arquivos e artefatos gerados durante a execução do projeto se encontram;
 - **Sprint backlog:** conjunto de funcionalidade, retiradas do *product backlog*, a serem desenvolvidas em uma *sprint*;
 - **Produto:** versão final ou parcial disponibilizado para o cliente no intuito de dar retorno do investimento no projeto;
 - **Ata:** registra a validação de uma implantação de funcionalidade;
 - **Planta de desenvolvimento:** conjunto de artefatos que descrevem o produto. Podendo usar os diagramas de: casos de uso, sequência, classes e entidade e relacionamento;
 - **Estrutura Analítica do Projeto (EAP):** organograma de apresentação do escopo do projeto;
 - **Cronograma:** organizar em sequencial os pacotes de trabalho dentro de um prazo determinado;
 - **Planilha de custo:** mapeia o custo de tempo gasto efetivamente na execução das atividades. Assim, é possível comparar o tempo final do projeto com o previsto com o objetivo de melhorar os processos proposto pelo Scrum Solo, agregando um objetivo do PSP.

O fluxo de desenvolvimento do Scrum Solo segue semelhante a metodologia Scrum, mas com pequenos detalhes.

Figura 3 – Fluxo do processo Scrum Solo



Fonte: (PAGOTTO et al., 2016)

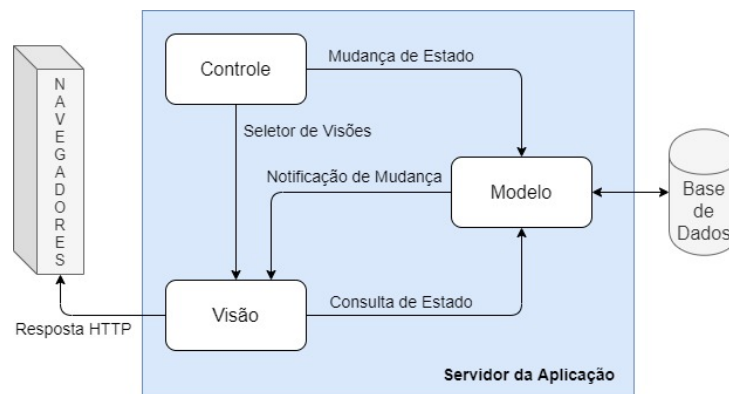
Conforme a Figura 3, o primeiro passo é estabelecer o *backlog* do produto. A partir daí, as atividades deste *backlog* são divididas para serem realizadas em várias *sprints*, definindo-se, então o *sprint backlog* para cada uma delas. É importante que o *sprint backlog* contenha funcionalidades exequíveis em uma semana, as quais são então validadas pelo cliente. Ao final das várias *sprints* o produto é concluído e colocado em produção.

Uma característica do Scrum Solo que difere do Scrum é que não ocorrem as reuniões diárias, já que o projeto consiste em somente um desenvolvedor. Ao final da *sprint* é exigida a disponibilização do produto ao cliente para que o grupo de validação possa validar e caso seja necessário ocorre a *oriented meeting*.

2.2.2 Arquitetura de Software MVC

Criada em 1979 por Trygve Reenskaug (JACYNTHO; SCHWABE; ROSSI, 2002), a arquitetura *Model-View-Controller* (MVC), em português Modelo-Visão-Controlador, propõe que o código seja organizado em três camadas, que são: modelo, visão e controle.

Figura 4 – A arquitetura MVC



Fonte: traduzido de (JACYNTHO; SCHWABE; ROSSI, 2002)

A camada de modelo é responsável por modelar a estrutura lógica da solução de um problema; esta estrutura é a única camada que é ligada diretamente à base de dados, observe a Figura 4. Já estrutura de visão é a camada de interação com o usuário, onde encontra-se a lógica de suporte ao usuário. A estrutura de controle é responsável pelo gerenciamento do negócio, preenchimento dos dados da estrutura de visão e requisição dos dados ao modelo, quando ocorre uma mudança de estado na aplicação. A arquitetura MVC foi adaptada e aplicada nas três aplicações WEB, API e *mobile* da ferramenta AaMT, o detalhe desta adaptação está descrito na seção 3.3.

2.2.3 Linguagens e Ferramentas de Desenvolvimento

Para desenvolvimento da aplicação aqui proposta escolheu-se ferramentas que possuem boa interoperabilidade. Em particular, o sistema proposto consiste em desenvolver uma aplicação *mobile* para os agentes que irão a campo coletar as informações e preencher dados de uma vistoria, uma aplicação WEB para exibição dos dados de forma detalhada e acompanhamento dos supervisores e coordenadores e uma API de comunicação com a base de dados. Assim, escolheu-se as seguintes tecnologias e linguagens: *React* (FACEBOOK, 2019a) para a aplicação WEB, *React-native* (FACEBOOK, 2019b) para a aplicação *mobile*, NodeJS (NODEJS, 2019) para a API, *Postgres* como base de dados das aplicações e a FIWARE para disponibilização dos dados abertos. Para a criação do artefato planta de *software*, descrito na seção 2.2.1.1, foi utilizado a notação *Unified Modeling Language* (UML).

A escolha das linguagens *React-native*, *React* e *nodeJS* deve-se ao fato de serem baseadas no JavaScript, mais especificamente as versões ES6+ do JavaScript. Desta forma, a utilização das linguagens em conjunto exigisse um menor esforço de codificação, já que são baseadas em JavaScript. A escolha da base de dados deve-se à familiaridade do autor com a tecnologia *Postgres* e a facilidade de instalar um servidor que suporte *Postgres*, facilitando assim a instalação da aplicação. Assim como o *Postgres*, optamos por usar a FIWARE por familiaridade do autor e

por ser possível realizar comunicação entre a API da AaMT e o componente ORION da FIWARE, para a disponibilização dos dados abertos.

Além das linguagens e base de dados para o desenvolvimento foi necessário o uso de ferramentas para testar e escrever o código. A ferramenta VisualCode (MICROSOFT, 2019) foi utilizada como ambiente integral de desenvolvimento para a escrita do código. A ferramenta Insomnia (INSOMNIA, 2019) foi também utilizada para testar as rotas de código da API desenvolvida.

Nas próximas seções as principais linguagens e ferramentas usadas para a construção da ferramenta AaMT são descritas.

2.2.3.1 React

React é uma biblioteca JavaScript criada pelo Facebook em 2011. Em 2013, o código foi disponibilizado para a comunidade ajudando a sua difusão entre desenvolvedores. Conforme descrito pelos criadores, o React torna a tarefa de criação de interface de usuário fácil.

Com o React é possível criar telas (chamadas de *views*) para os estados da aplicação e, à medida que as informações forem mudando, o React se responsabiliza por atualizar somente os componentes necessários, de acordo com a mudança de estado da aplicação. É possível também criar componentes encapsulados com seus próprios estados que podem ser combinados em uma Interface de Usuário.

Esta biblioteca foi utilizada para o desenvolvimento da *interface* WEB da ferramenta AaMT e comunicação com a API do projeto.

2.2.3.1.1 React Native

O React Native é a aplicação do React no mundo *mobile*, também criada pelo Facebook. Esta biblioteca faz a interpretação do código React Native tanto para Objective C/Swift para iOS ou Java/Kotlin para o Android. A vantagem desta interpretação é que o desenvolvedor não precisa construir a aplicação duas vezes para plataformas diferentes.

Além disto, como o React Native é uma biblioteca JavaScript, esta é compatível com quase toda biblioteca JavaScript. Essa característica permite que sejam reaproveitados muitos códigos usados na aplicação WEB da AaMT, como por exemplo as requisições realizadas à API.

2.2.3.2 Node.js

Criado em 2019, o Node.js usa o mecanismo JavaScript V8, tecnologia do navegador Google Chrome que compila a linguagem fora do navegador.

Assim como a linguagem JavaScript, o Node.js é assíncrono. Uma das suas peculiaridades é que, ao receber uma requisição e esta necessitar fazer um acesso bloqueante, como uma

requisição à base de dados, o Node.js não bloqueia a aplicação, ele continua a execução com outras requisições e quando o processo finalizar, ele retoma a tarefa anterior. Essa característica permite que o Node.js faça bom proveito do uso da CPU e manipule múltiplas conexões com um único servidor. Além disto, suas bibliotecas são gravadas usando o paradigma não-bloqueadores. Assim, o uso de bloqueadores é usado somente quando necessário na aplicação.

Além das vantagens citadas acima, usar Node.js é contar com uma vasta biblioteca contida nos repositórios da *Node Package Manager*, Gerenciador de Pacotes do Node, que possui mais de um milhão de pacotes de código aberto. Os principais destes pacotes utilizados no desenvolvimento da AaMT foram:

- **Express:** *framework* para aplicativos Node.js com recursos para métodos HTTP e *middlewares*. É uma biblioteca que possibilita a criação de API usando Node.js, ([STRONGLOOP, 2020](#));
- **Sequelize:** é um *Object-Relational Mapping*, mapeamento objeto-relacional, para *Postgres*, *MySQL*, *MariaDB*, *SQLite* e *Microsoft SQL Sever*. É um *framework* para aplicativos Node.js, ([SEQUELIZE, 2020](#)).

2.2.3.3 A plataforma FIWARE

Criada pela Comissão Europeia, o objetivo da plataforma FIWARE é produzir um ecossistema sustentável e totalmente aberto para fomentar a criação de de aplicações verdadeiramente inteligentes, de código aberto, livre de *royalties* e sem a dependência de fornecedores de tecnologias. A FIWARE é composta por sete componentes ([FIWARE, 2019](#)):

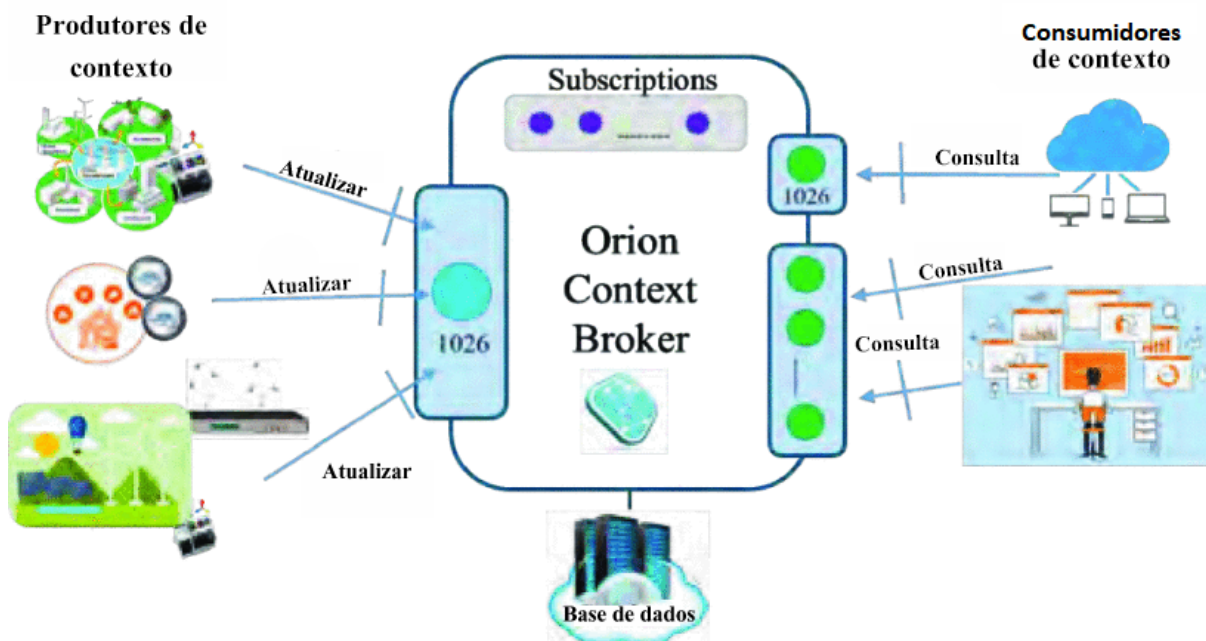
- **ORION Context Broker:** principal elemento da FIWARE. Toda aplicação baseada na plataforma contém a implementação do ORION. Neste componente é feito o registro das informações de contexto, permitindo a criação de aplicações produtoras e consumidoras de contexto;
- **Conexão à Internet of Things (IoT):** agentes, chamados de IDAS IoT, são capazes de produzir/consumir informações de contexto existentes no meio físico;
- **Autorização de manuseio e controle de acesso às APIs:** estruturas para configuração de acesso e autorização, baseadas em padrões de segurança como: OAuth e *eXtensible Access Control Markup Language*;
- **Publicação de informações de contexto como dados abertos:** a FIWARE incorpora o CKAN como parte da arquitetura, ferramenta para criação de portais de dados abertos ([CKAN, 2019](#));

- **Grande análise de dados de informação de contexto histórico:** incorpora o software Cygnus como parte da arquitetura, assim como o CKAN, que permite armazenar dados históricos de informações de contexto, permitindo análise de dados (*Big Data*) ou consultas avançadas sobre as informações históricas;
- **Painéis de aplicações:** visando facilitar a criação de aplicativos, sem o conhecimento de desenvolvimento, a FIWARE disponibiliza a ferramenta *Wirecloud* em seu ambiente de laboratório (FIWARE Lab);
- **Processamento em tempo real de fluxos de mídia:** faz uso da ferramenta Kurento para permitir o processamento de multimídia para incorporar recursos de detecção baseado em mídias como fotos, áudios e vídeos.

Além destes componentes, a FIWARE provê *Generic Enablers* (GE), os habilitadores genéricos possuem, que possuem funcionalidades de âmbito geral como serviços de hospedagem em nuvem, gerenciamento de dados, IoT, etc. É possível desenvolver aplicações específicas somente fazendo a interação entre os GEs. Além dos GEs, a FIWARE fornece também os *Domain Specific Enablers*. Eles possuem características de desenvolvimento iguais aos GEs, porém são direcionadas para resolver problemas de domínios específicos da população, como saúde, educação, etc.

Neste trabalho o componente ORION foi utilizado para recebimento dos dados autorizados coletados nas vistorias de campo. Todo GE desenvolvido com a FIWARE faz a implementação do ORION. Este é o componente que faz a comunicação, por meio de API, entre aplicações consumidoras, as aplicações produtoras e a base de dados.

Figura 5 – Orion Context Broker



Fonte: traduzido de (PLAMANESCU et al., 2019)

Como mostra a [Figura 5](#), as aplicações podem realizar atualização, caso sejam autorizadas, ou consultar. As aplicações que realizarão atualização de dados são chamadas de produtores de contexto e as que consultam são chamadas de consumidores de contexto. Por padrão o ORION utiliza a porta de acesso de rede 1026, tanto para atualização como para consulta. Além disso, o ORION possui a tecnologia de *subscribe*. Esta tecnologia permite que consumidores de contexto se registrem no servidor que contém ORION e quando um dado for alterado, na base, o ORION emite uma notificação de mudança de contexto às aplicações inscritas.

Essas requisições feitas ao ORION usam o protocolo HTTP e para obter uma comunicação de sucesso com o componente devem obedecer o padrão NGSI versão dois, padrão desenvolvido pela própria FIWARE. Um exemplo de requisições HTTP de acordo com o padrão NGSI versão dois, usando a biblioteca jQuery, podem ser vistas no [Apêndice B](#).

O ORION da FIWARE será utilizado para armazenar os dados autorizados, das vistorias coletadas em campo pelo usuários da ferramenta AaMT, que permitam a disponibilização em larga escala como dados abertos para a comunidade científica, entidades governamentais, etc. A ferramenta propõe que os dados relevantes como a pesquisa de campo, apresentada nos formulários dos [Anexo A](#) e [Anexo B](#), os resultados das amostras e índices como o IB, IP e ITR sejam enviados para o ORION em formato de JSON para que seja acessado por meio de API.

2.2.3.4 Unified Modeling Language - UML

A *Unified Modeling Language* (FOWLER, 2005), no português Linguagem de Modelagem Unificada, foi criada por Grady Booch, James Rumbaugh e Ivan Jacobson, em uma tentativa

de unificar diversas técnicas de modelagem Orientadas a Objeto (OO) de forma que qualquer sistema possa ser modelado corretamente, sendo compreensível, editável, escalável, de fácil acoplamento com outros softwares, etc.

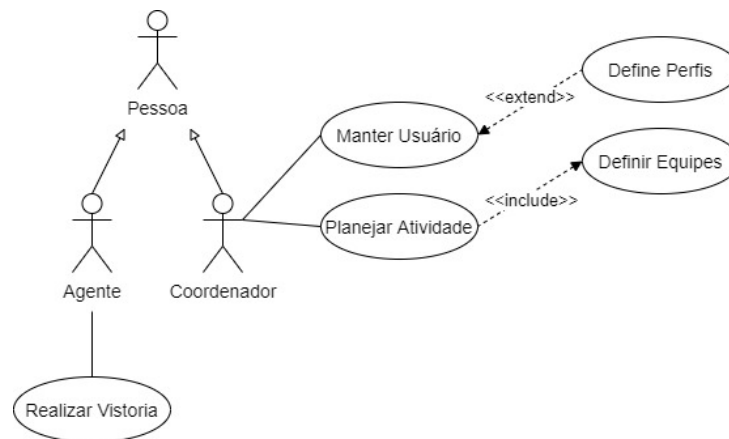
A UML foi utilizada neste trabalho para o desenvolvimento dos artefatos gerados na execução das atividades da metodologia Scrum Solo. As seções a seguir apresentam os diagramas usados na criação dos artefatos necessários à execução das atividades de desenvolvimento deste projeto.

2.2.3.4.1 Diagrama de Caso de Uso

Este diagrama descreve os requisitos funcionais de um sistema sobre a perspectiva do usuário final do sistema. O diagrama é descrito por atores, casos de uso e o sistema modelado. A Figura 6 ilustra os vários elementos do diagrama a seguir descritos.

- **Ator:** representa relações de entidades com o sistema como usuários, clientes, outros sistemas, etc. Atores são representados por bonecos;
- **Caso de uso:** representa o cenário típico de um requisito funcional. É possível representar o cenário alternativo de um caso de uso, se necessário. Casos de uso são representados por elipses;
- **Relações:**
 - **Associação:** uma associação pode existir entre um ator e um caso de uso. São representados por uma linha reta;
 - **Generalização e Especialização:** forma de associação na qual existem características semelhantes entre os casos de uso (generalização) ou entre os atores (especialização). São representadas por uma seta;
 - **Inclusão:** também chamados de subrotinas, significa que a execução de um caso de uso obriga, também, sua execução. São representado por uma seta pontilhada com a descrição no meio dela da seguinte forma "*«include»*";
 - **Extensão:** Encapsula um cenário alternativo complexo em um caso de uso alternativo. Na descrição escrita do caso de uso deve ser preenchido o campo "gatilho" na descrição do caso de uso para indicar quando a extensão deve entrar em ação. Assim como a inclusão, são representados por uma seta pontilhada porém a contém "*«extend»*".

Figura 6 – Exemplo de um diagrama de casos de uso



Fonte: os autores

A Figura 6 é uma variação do caso de uso real da ferramenta AaMT para facilitar a explicação dos elementos do diagrama.

2.2.3.4.2 Diagrama de Classes

Os diagramas de classes modelam a estrutura estática das classes de um sistema. É possível usar o diagrama de classe para modelar os objetos, exibir suas relações e descrever os serviços que oferecem, (IBM, 2016).

No diagrama de classes existe o conceito de classe e objeto. Uma classe contém um agrupamento de objetos, sua estrutura representa a abstração de uma entidade do mundo real e um objeto é uma entidade do mundo real. Por exemplo, pode-se dizer que animal é uma classe e um cachorro é um objeto da classe animal.

A notação UML, divide a estrutura da classe em três: identificação da classe, atributos e métodos, como mostra a Figura 7. Os atributos representam características da classe e os métodos são processos realizados pelos objetos que representam seu funcionamento, por exemplo: pode-se considerar o latir de um cachorro um método da classe animal.

Figura 7 – Exemplo de uma classe



Fonte: os autores

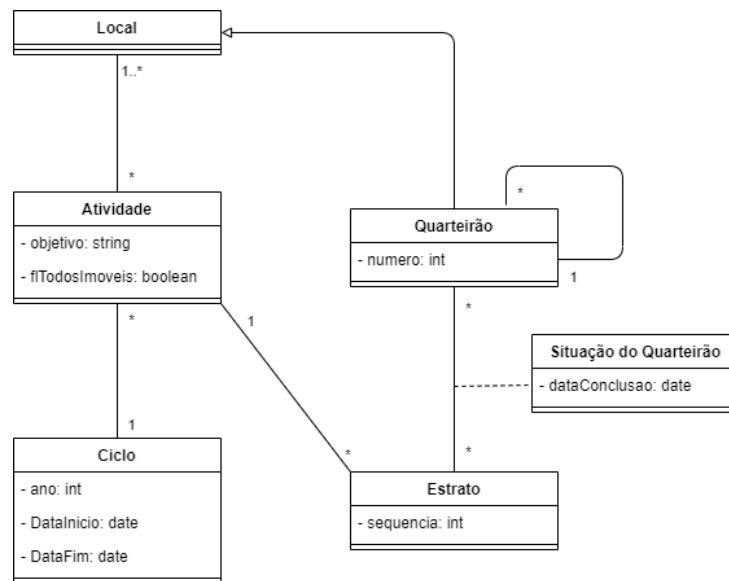
Na Figura 7 podemos notar que a classe Ciclo não possui métodos e contém três atributos com nome e tipo. Os atributos possuem permissões de acesso que são definidos pelos sinais antes de seus nomes, que são:

- + : visibilidade pública, pode ser acessado pelo exterior da classe;
- - : visibilidade privada, somente a própria classe pode acessar o atributo;
- # : visibilidade protegida, é acessível por classes derivadas.

Uma única classe não consegue modelar problemas mais complexos do mundo real. Assim a modelagem geralmente envolve um conjunto de classes que se relacionam. Existem múltiplas associações e cada uma delas possui uma multiplicidade em cada ponta. A Figura 8 ilustra associações entre classes, que são:

- **Associação Normal:** representa uma conexão entre classes de forma mais comum. Essa associação é representada por uma linha entre classes;
- **Associação Recursiva:** representa a mesma conexão que uma normal, porém a conexão se dá com a mesma classe. Essa associação é representada por uma linha com as duas extremidades ligadas à mesma classe;
- **Associação de Classe:** serve para associar uma classe a uma associação para adicionar informações extras na relação entre outras classes. Essa associação é representada por uma classe com uma linha pontilhada ligada a outra associação;
- **Agregação:** indica que uma das classes está contida ou é uma parte da outra classe. Esta associação contém agregação ou composição. Uma agregação normal indica que uma classe está contida na outra, podendo inclusive estar contida várias vezes na outra. Já a composição é associada a um único objeto da outra classe, não podendo se relacionar mais de um objeto da mesma classe. Essa associação é representada por uma linha com um quadrado na ponta, quadrado branco para agregação e preto para composição;
- **Generalização Normal:** nesta generalização a subclasse herda as características da classe geral, chamada de superclasse. Essa associação é representada por uma seta.

Figura 8 – Exemplo de associações entre classes



Fonte: os autores

A Figura 8 ilustra uma associação normal entre as classes Ciclo e Atividade e uma associação recursiva em Quarteirão. Além disto, a classe Quarteirão também é uma generalização de Local e possui uma relação com Estrato, esta relação, por sua vez, contém uma associação de classe, como é possível ver na classe Situação do Quarteirão.

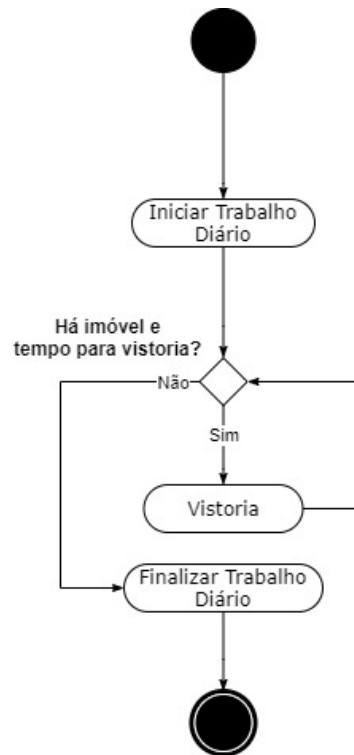
2.2.3.4.3 Diagrama de atividades

Os diagramas de atividades representam o fluxo das atividades de uma regra de negócio. Este diagrama serve para demonstrar, de forma ilustrativa, como é o comportamento de um conjunto de atividades para a compreensão da solução aplicada em cima da regra de negócio (FOWLER, 2005). A Figura 9 ilustra os elementos básicos da construção de um diagrama de classes, que são:

- **Nó inicial:** início do processo. É representado por um círculo preenchido de preto;
- **Fluxo ou Aresta:** indica o sentido do fluxo. É representado por uma seta;
- **Partição:** indica quem executa as ações contidas na partição. é representado por um retângulo, horizontal ou vertical, com um cabeçalho;
- **Ação:** atividade que deve ser executada antes de partir para o próximo fluxo. É representado por um retângulo com bordas arredondada;
- **Decisão ou Desvio:** ponto de tomada de decisão do fluxo, contém um único fluxo de entrada e vários fluxos de saída. É representado por um losango;

- **Nó final:** indica o fim do fluxo, em um diagrama de atividades pode haver mais de um nó final, porém somente um nó inicial. É representado por um círculo preenchido de preto e circulado por outro círculo.

Figura 9 – Exemplo de diagrama de atividades



Fonte: os autores

2.3 Trabalhos Relacionados

A luta contra doenças endêmicas é um problema mundial e desenvolver um software que seja capaz de atender todas as regras de negócio do combate a estas doenças requer muito esforço de análise e desenvolvimneto. Por isso, existem muitos trabalhos que abordam a problemática com soluções específicas.

Uma aplicação voltada para o mosquito *Aedes aegypti* é a Dengui-MI (EIRAS; RESENDE, 2009). Este sistema possui armadilhas informatizadas para capturar fêmeas grávida do *Aedes aegypti*, que fazem transmissão dos dados em tempo real. O dados coletados pelas armadilhas são expostos em mapas georreferenciados. O objetivo da ferramenta é levantar índices para calcular o nível de criticidade das regiões e auxiliar o combate ao mosquito *Aedes aegypti*.

Em uma parceria entre universidades e a Secretaria Municipal de Saúde do Rio de Janeiro (CODECO et al., 2018) foi desenvolvido o InfoDengue. O sistema coleta dados de redes sociais, casos notificados de Dengue, dados meteorológicos e dados demográficos dos municípios

brasileiros. Seu objetivo é calcular níveis de alertas, com base em cálculos estatísticos, que são baixa transmissão, atenção, transmissão e alta incidência.

No combate contra o mosquito *Aedes aegypti* pode-se destacar o uso de larvicida. Segundo estudos do MS, populações do mosquito *Aedes aegypti* pode criar resistência a larvicida (MS, 2013). A ferramenta MIRO foi criada com o objetivo de mapear populações que desenvolveram resistência a larvicidas com dados de todo o mundo, (DIALYNAS et al., 2009).

A ferramenta proposta por Hamer e outros (HAMER; CURTIS-ROBLES; HAMER, 2018) busca suprir a necessidade de dados dos pesquisadores de arboviroses com o auxílio da população. A ideia é fazer uso da ciência cidadã, disseminando o aplicativo para que a população responda questionários e envie os dados para uma base de dados acessível aos pesquisadores.

Já Valero e outros (VALERO et al., 2018) criaram um aplicativo *mobile* OlympTRIP para monitorar a saúde dos atletas olímpicos espanhóis que vieram para os jogos olímpicos no Rio de Janeiro em 2016. O motivo da criação do aplicativo foi evitar a infestação de arboviroses através dos atletas participantes dos jogos.

A qualidade de um software não está ligada somente ao poder de realização de tarefas de um software, mas também a sua escalabilidade. Alcançar esta meta não é trivial e requer o uso de linguagens e ferramentas adequadas para a solução do problema. Fazer o uso da FIWARE como ferramenta para disponibilização de dados de interesse populacional vai além do suporte tecnológico gerado por seus componentes. A FIWARE possibilita a criação de agentes da IA capazes de trabalhar sobre os dados gerados pelos ACEs em tempo real, fazendo uso da ferramenta de publish/subscribe disponibilizada no componente ORION da FIWARE.

Um exemplo do uso do subscribe é o aplicativo na área *e-health* (GALÁN et al., 2015), que faz monitoramento remoto de paciente para avaliação de saúde em tempo real com o uso da plataforma FIWARE. O objetivo da aplicação é melhorar a assistência remota dos pacientes que estão em casa, com o intuito de melhorar o fluxo de trabalho dos médicos.

Outro exemplo do uso da FIWARE na área de saúde é aplicação (RUIZ et al., 2018) de monitoramento ambiental onde os autores desenvolveram um sistema embarcado que faz uso do ORION para analisar poeiras em ambientes de tratamento intensivo. O objetivo de identificar a concentração das partículas e determinar o grau de risco de propagação de doenças.

No desenvolvimento da ferramenta AaMT foi feito um estudo aprofundado de como é realizado o combate ao mosquito *Aedes aegypti*, além de entrevistas com especialistas na área, como a Dra. Roseli La Corte dos Santos. Portanto, diferente das ferramentas anteriormente citadas, a ferramenta AaMT é uma aplicação que busca gerir todo o processo de combate ao mosquito, desde o planejamento e apoio ao estudo de ciclos epidemiológicos até o suporte automático às atividades de vistorias pelo ACEs.

3

Proposta da Ferramenta AaMT

Para o desenvolvimento do projeto e protótipo da ferramenta AaMT foi utilizada a metodologia Scrum Solo. De acordo com esta metodologia o passo inicial é a definição dos responsáveis pelos papéis que comporão a equipe de desenvolvimento e o repositório onde ficarão os artefatos. Para a consecução do projeto foram definidos:

- *Product owner*: Sidney Lourdes César Souza Sa, José Oliveira Santos, agentes da Secretaria de Estado da Saúde de Sergipe, e a Dra. Roseli La Corte dos Santos, pessoas a quem o *software* está destinado a atender nesta primeira versão;
- Orientadores: papéis assumidos pelas professoras Dra. Leila Silva e Dra. Débora Nascimento;
- Desenvolvedor: Weslan Rezende Alves;
- Grupo de validação: não houve validação do protótipo, por isso, não foi necessário grupo de validação;
- Repositório: o repositório de arquivos disponível em nuvem pode ser acessado pelo link https://drive.google.com/drive/folders/1vgprd4jyFaIg1Mbbr9y94l2muAy_eKBt?usp=sharing;
- Repositório do código fonte: o código fonte foi posto no github para desenvolvimento e versionamento de código, acessível pelo link <https://github.com/DCOMP-UFS/AAMT>.

De acordo com o fluxo expresso na [Figura 3](#), a primeira etapa do ciclo de desenvolvimento foi a construção do *product backlog*. No entanto, para gerar este artefato faz-se necessário o levantamento de requisitos, descrito na [seção 3.1](#). Após esta etapa o *product backlog* foi dividido em *sprints* para ser desenvolvidas. O protótipo da ferramenta contou com seis *sprints* e a descrição

do *product backlog* e de algumas destas *sprints* é o objeto da [seção 3.1](#). O Scrum Solo é iterativo e incremental. O projeto do sistema foi assim construído incrementalmente ao longo das seis *sprints*, obtendo, ao final do projeto, a arquitetura expressa na [seção 3.3](#). O fluxo típico de uma seção da ferramenta pode ser expresso pelo diagrama de atividades, descrito na [seção 3.3.2](#). As principais funcionalidades do sistema são ilustradas pelas telas da [seção 3.4](#).

3.1 Levantamento de Requisitos

A elicitação dos requisitos foi feita na atividade *requirement*, conforme a metodologia Scrum Solo, através de visita técnica, em Simões Dias, e reuniões com os orientadores e os *product owners* do projeto. Para a validação dos requisitos foi necessário uma reunião de apresentação de pré-protótipo. Os requisitos elicitados são apresentados nos quadros a seguir.

Quadro 1 – Requisitos não funcionais da ferramenta AaMT

Código	Requisito Não Funcional	Descrição
RNF001	Preenchimento ágil de boletim	A aplicação <i>mobile</i> da ferramenta deve prover meios de facilitar o preenchimento dos formulários de vistoria, a fim de dar maior produtividade em campo.
RF002	Facilidade em geração de relatório	O sistema deve produzir relatórios com poucos cliques e, se possível, possibilitar a geração de relatórios customizados.
RF003	Realizar vistorias offline	O sistema deve dispor da opção de vistoria mesmo que não haja internet.
RF004	Autenticidade dos dados do usuário	O sistema deve conter uma tela de login para garantir que a informação é proveniente da fonte anunciada e que não foi alvo de mutações.
RF005	Privacidade dos dados abertos	O sistema deve disponibilizar como dados abertos somente os dados autorizados pelos coordenadores gerais e em comum acordo com a Lei Geral de Proteção de Dados (lei 13.709/18).

Fonte: os autores

Quadro 2 – Requisitos funcionais da ferramenta AaMT

Código	Requisito Funcional	Descrição
RF001	Definir Ciclo	Organizar as atividades executadas em ciclos para refinamento de relatórios.
RF002	Manter Município	O coordenador geral pode cadastrar, alterar, consultar ou excluir os municípios da sua regional de saúde.
RF003	Manter Usuário	O coordenador geral gerencia todos os perfis, enquanto o coordenador municipal pode gerir apenas agentes, supervisores e laboratoristas.
RF004	Manter Localidade/Bairro	Os coordenadores e supervisores podem cadastrar, alterar, consultar ou excluir as localidades e bairros do município.
RF005	Manter Zona	O coordenador municipal pode cadastrar, alterar, consultar ou excluir zonas. As zonas são um conjunto de quarteirões e servem para refinamento de relatórios.
RF006	Manter Atividade	O coordenador geral pode criar atividades, do LIRAa ou PNCD, para um ou mais município pertencentes à sua regional de saúde, enquanto o coordenador municipal pode criar atividades somente para seu município.
RF007	Manter Laboratório	O coordenador municipal é responsável por cadastrar os laboratórios existente em seu município.
RF008	Planejar Atividade	O coordenador municipal deve planejar os estratos e equipes que irão trabalhar na atividade.
RF009	Manter Rua	O supervisor pode cadastrar, alterar, consultar e excluir as ruas do município.
RF010	Manter Quarteirão	O supervisor pode cadastrar, alterar, consultar e excluir os quarteirões do município.
RF011	Manter Imóvel	O supervisor pode cadastrar, alterar, consultar e excluir os imóveis do município.
RF012	Iniciar Trabalho Diário	O agente antes de ir a campo define os quarteirões e os lados que irá trabalhar no dia.
RF013	Finalizar Trabalho Diário	Processo de sincronização das informações offline, registradas nas vistorias, com o servidor através da API da ferramenta AaMT ao mesmo tempo a API transmite os dados aberto para a FIWARE.
RF014	Realizar Vistoria	O agente pode cadastrar as vistorias realizadas em imóveis.
RF015	Realizar Exame	O laboratorista pode cadastrar, alterar, consultar os exames das amostras coletadas em campo.
RF016	Enviar dados para FIWARE	Os dados das vistorias de campo, resultados das amostras e os índices dos municípios devem serem disponibilizados como dados abertos, pela FIWARE.
RF017	Relatório diário	Emite o relatório com as informações coletadas no dia de trabalho de um ou mais agentes.
RF018	Relatório semanal	Emite o relatório com as informações coletadas durante a semana epidemiológica.
RF019	Manter Regional de Saúde	O coordenador geral é responsável por manter os regionais de saúde.
RF020	Relatório por Ciclo	Os coordenadores gerais e municípios pode visualizar os relatórios por ciclos.
RF021	Resultados do ciclo	Coordenador geral pode consultar os dados coletados durante um ciclo em todos os municípios.
RF022	Resultados do atividade	O coordenador geral pode visualizar os resultados de uma atividade.
RF023	Resumo do trabalho diário	Um coordenador municipal pode visualizar o resumo do trabalho diário realizado em seu município.
RF024	Resumo do resultado das amostras	Coordenador municipal pode acompanhar o resumo dos resultados das amostras que foram examinadas, um supervisor só pode acompanhar os resultados das amostras de suas equipes.

RF025	Resultados do atividade por município	O coordenador municipal pode visualizar o resultado de uma atividade realizada em seu município.
RF026	Acompanhamento de demandas dos laboratórios	O coordenador municipal pode acompanhar as demandas das amostras por laboratório, já um supervisor pode acompanhar apenas suas demandas.
RF027	Resumo do trabalho diário por equipe	O coordenador municipal pode visualizar o resumo do trabalho diário por equipe, enquanto o supervisor só pode visualizar o resumo das suas equipes.
RF028	Visualizar boletim de campo	Supervisores e coordenadores municipais podem visualizar os registros do boletim de campo de um agente. Um agente só pode visualizar os seus boletins de campo.
RF029	Encaminhar amostras	O supervisor pode encaminhar amostras para um laboratório examina-las.
RF030	Definir trabalho diário	O supervisor é responsável por definir as rotas de cada agente de sua equipe antes de irem a campo.

Fonte: os autores

Após a etapa de *requirements* da metodologia Scrum Solo, foi realizado o planejamento do protótipo da ferramenta AaMT e o diagrama de casos de uso, antes de estabelecer o *backlog* do produto e iniciar as *sprints*. Analisando o diagrama casos de uso descrito na [Figura 10](#), foi possível identificar quatro casos de uso críticos que são: Definir Ciclo, Manter Atividade, Planejar Atividade, Definir Trabalho Diário, Iniciar Trabalho Diário, Finalizar Trabalho Diário, Realizar Vistoria e Realizar Exame.

Figura 10 – Diagrama de Caso de Uso



Fonte: os autores

Além dos casos de uso críticos, a [Figura 10](#) apresenta quatro atores, aos quais foram atribuídas as seguintes responsabilidades:

1. **Coordenador Geral:** generalização de um usuário responsável por gerenciar os ciclos, as atividades executadas, os usuários e as regionais de saúde do sistema;
2. **Coordenador:** generalização de usuário responsável por gerir os usuários do seu município, criar atividades municipais e planejar as atividades de responsabilidade do município;
3. **Supervisor:** generalização de usuário responsável por manter as localidades, imóveis e ruas do município. Encaminhar as amostradas coletadas para os laboratórios e, além disto, este usuário é capaz de cadastrar boletins de campo em caso que o agente não foi capaz de realizar ou não tem habilidades para o mesmo;

4. **Agente:** generalização de um usuário responsável por coletar as informações do sistema em campo. Este ator representa a parte executora do monitoramento, voltada para a coleta de dados e amostras que servem como subsídio para o cálculo dos indicadores de epidemias;
5. **Laboratorista:** generalização de um usuário responsável por examinar as amostras coletadas pelos agentes durante as vistorias.

Um fator importante a destacar da ferramenta em relação aos usuários é que a AaMT pode gerir mais de um município e, para isto, foi feito o controle de localidade da seguinte forma:

- **Regional de Saúde:** localidade que engloba vários municípios e pertence a um estado. Este local é gerido por um Coordenador Geral, o qual é responsável por cadastrar outras regionais de saúde e seus coordenadores gerais;
- **Município:** região que pertence a uma regional saúde. Para a aplicação um município deve executar as atividades impostas pelo regional de saúde. A gestão do município é dada ao Coordenador, este também pode executar outras atividades, além da proposta pelo seu coordenador geral, de acordo com a situação do município.

A descrição de todos os casos de uso se encontra no [Apêndice A](#). Nas próximas seções iremos detalhar os casos de uso críticos, para a descrição dos casos de uso adotamos um modelo de acordo com a notação UML.

3.1.1 Definir Ciclo

Um ciclo representa um período de tempo no ano. Normalmente este período é dado por dois meses, contendo assim seis ciclos em um ano. Porém, existem municípios com pouca mão de obra, que não são capazes de realizar atividades no período de um ciclo.

Para estes casos, optou-se por modelar o ciclo maleável, ficando a cargo do Coordenador Geral definir o período do ciclo adequado para a sua regional de saúde. Este caso de uso é descrito como segue:

- **Pré-condições:** o coordenador geral está logado e com *token* de validação armazenado;
- **Pós-condições:** os dados do ciclo foram armazenados.
- **Cenário principal:**
 1. O coordenador geral escolhe cadastrar um ciclo;
 2. O sistema exibe tela de cadastro de ciclo;
 3. O coordenador geral define a data de início e fim do ciclo, a sequência do ciclo e as atividades que deseja realizar em todos os municípios naquele ciclo;

4. O sistema salva as informações e retorna mensagem de sucesso ao coordenador geral.

- **Fluxos Alternativos:**

- 4a. A data final do ciclo é menor que a data inicial;
 1. O Sistema informa que a data fim é menor que a data início;
- 4b. O período informado do novo ciclo já está contido no período de outro ciclo já cadastrado.
 1. O Sistema informa que já existe outro ciclo contendo o mesmo período informado;

3.1.2 Manter Atividade

Uma atividade é consituída por metodologia e objetivos, por exemplo, é possível realizar um levantamento de índice mais tratamento com a metodologia PNCD. Isto significa que o formulário de vistoria é diferente para cada atividade ou que nem todos imóveis podem ser vistoriados, conforme foi visto na metodologia LIRAA, na [seção 2.1.2](#). Este caso de uso é descrito como segue:

- **Pré-condições:** o coordenador geral esta logado e com *token* de validação armazenado;
- **Pós-condições:** a atividade foi registrada e armazenada.
- **Cenário principal:**
 1. O usuário escolhe cadastrar atividade;
 2. O sistema exibe tela e cadastro de atividade;
 3. O usuário define o objetivo da atividade, a metodologia, o objetivo da metodologia, a abrangência da atividade, se a execução será em todos os imóveis ou não, o ciclo e os municípios que a atividade será vinculada;
 4. O Sistema retorna ao usuário a lista de todas atividades e informa o cadastro da atividade com sucesso.

3.1.3 Planejar Atividade

Antes dos agentes irem a campo, os coordenadores devem planejar a atividade que será executada. O planejamento consistem em planejar os estratos e equipes que irão executar a atividade.

- **Pré-condições:** o coordenador está logado e com o *token* de validação armazenado;
- **Pós-condições:** os dados de planejamento foram armazenados.

- **Cenário principal:**

1. O coordenador acessa página de todas as atividades;
2. O sistema exibe a lista de atividades do município por ciclo;
3. O coordenador escolhe uma atividade ainda não planejada;
4. O sistema exibe tela de planejamento de atividade;
5. O coordenador escolhe os locais do estrato e cadastrar o estrato;
6. O sistema adiciona o estrato a lista de estratos planejados;
7. O coordenador escolhe os membros de uma equipe, os locais, o supervisor e cadastra a equipe;
8. O sistema adiciona a equipe a lista de equipes planejadas;
9. O coordenador finaliza o planejamento da atividade;
10. O sistema salva as informações e retorna mensagem de sucesso ao coordenador.

- **Fluxos Alternativos:**

- 7a. O coordenador cadastra um novo estrato;
 1. retorna para o passo 5;
- 9a. O coordenador cadastra uma nova equipe;
 1. retorna para o passo 7;

3.1.4 Definir Trabalho Diário

Este caso de uso representa a atividade de planejamento das rotas que os agentes irão vistoriar no dia. Essa atividade é realizada pelos supervisores antes dos agentes irem a campo.

- **Pré-condições:** o supervisor está logado e com o *token* de validação armazenado;

- **Pós-condições:** registro de novo trabalho diário na base.

- **Cenário principal:**

1. O supervisor escolhe uma atividade;
2. O sistema exibe as equipes de responsabilidade do supervisor;
3. O supervisor escolhe definir trabalho diário;
4. O sistema leva a tela de definição;
5. O supervisor escolhe a equipe, o agente, os quarteirões e lados do agente;
6. O sistema registra o trabalho diário e pergunta se o supervisor deseja definir novo trabalho diário;

7. O supervisor escolhe não;
8. O sistema emite mensagem de sucesso;

- **Fluxos Alternativos:**

- 7a. Supervisor escolhe sim;
 1. Retorna ao passo 4;

3.1.5 Iniciar Trabalho Diário

Este caso de uso é realizado pelos agentes antes de irem a campo para preparar o aparelho móvel para o trabalho offline.

- **Pré-condições:** o agente está com um aparelho móvel, logado, com o *token* de validação armazenado e o trabalho diário planejado pelo supervisor;
- **Pós-condições:** o registro da hora de início do trabalho e dados atualizados da base baixadas no aparelho móvel.
- **Cenário principal:**
 1. O agente escolhe a atividade;
 2. O sistema consulta a equipe, do agente na atividade, e os quarteirões da equipe;
 3. O agente clica em iniciar trabalho diário;
 4. O sistema consulta os quarteirões e lados associados ao trabalho diário para o agente no dia e atividade escolhida e faz o *download* dos quarteirões e os imóveis pertencentes para o cache do aparelho móvel;
 5. O sistema registra o início de trabalho diário e retorna mensagem de sucesso ao agente.
- **Fluxos Alternativos:**
 - 4a. O supervisor não definiu o trabalho diário para o agente;
 1. O sistema informa que não há trabalho diário definidor pelo supervisor e retorna ao passo 1;

3.1.6 Finalizar Trabalho Diário

Após o fim de um trabalho de campo, o agente deve realizar o processo de finalizar o trabalho diário para sincronizar as informações coletadas durante o trabalho de campo com o servidor.

- **Pré-condições:** o agente está com um aparelho móvel com acesso a internet, logado e com o *token* de validação armazenado;
- **Pós-condições:** os dados das vistorias foram armazenados.
- **Cenário principal:**
 1. O agente encerra os trabalhos do dia;
 2. O sistema acessa as informações em *cache*, salva as informações e informa mensagem de sucesso ao agente.

3.1.7 Realizar Vistoria

Este caso de uso ocorre quando os ACEs vão a campo realizando as atividades nos imóveis planejados para execução da mesma. Este caso de uso é descrito como segue:

- **Pré-condições:** o agente ou supervisor está logado com *token* de validação armazenado e com o trabalho diário iniciado;
- **Pós-condições:** vistoria registrada em *cache*;
- **Cenário principal:**
 1. O usuário escolhe registrar nova vistoria;
 2. O Sistema carrega os campos do formulário;
 3. O usuário busca o imóvel da vistoria e preenche os campos solicitados;
 4. O Sistema registra as informações em *cache* e emite mensagem de vistoria armazenada em *cache*;
- **Fluxos Alternativos:**
 - 3a. O agente não encontra o imóvel cadastrado nas informações em *cache*;
 1. O sistema informa que o imóvel não está registrado na base;
 2. O agente registra os dados do imóvel;
 3. O sistema salva as informações em *cache* para sincronização com servidor durante a finalização do trabalho diário;
 - 3b. A atividade executada é PNCD com e contém o objetivo de tratamento;
 1. O sistema solicita o preenchimento do formulário de tratamento;
 2. O agente preenche o formulário;
 - 3c. Registro de uma amostra no imóvel;
 1. O agente solicita registro de amostra;
 2. O sistema exibe uma caixa flutuante com formulário de registro de amostra;
 3. O agente preenche o formulário;

3.1.8 Realizar Exame

Após atividade de campo as amostras são encaminhadas aos laboratórios para exames e os resultados devem constar no sistema para que seja possível a emissão de alguns relatórios. Este caso de uso é descrito como segue:

- **Pré-condições:** o laboratorista está logado com *token* de validação armazenado;
- **Pós-condições:** exame registrado no sistema referente a sua amostra;
- **Cenário principal:**
 1. O usuário escolhe a amostra e informa o código ao sistema;
 2. O Sistema identifica a amostra e retorna os dados e o formulário de exame;
 3. O usuário preenche os campos de acordo com os resultados dos exames;
 4. O Sistema registra as informações emite mensagem de exame cadastrado com sucesso;
- **Fluxos Alternativos:**
 - 4a. O usuário realiza o registro de exames de outra amostra;
 1. Retorna ao passo 1;

3.2 Backlog do Produto e Sprints

Um dos principais artefatos da metodologia Scrum Solo é o *product backlog* provinda da tarefa *requirement*. Cada requisito da ferramenta acrescentou um item ou um conjunto de itens ao *product backlog* descritos na [Quadro 3](#). O termo manter nos requisitos determinam o conjunto de ações: cadastrar, leitura, alterar e excluir.

O *product backlog* segue o modelo disponibilizado pela Scrum Solo, nas colunas temos o id, descrição, data de inserção e data de seleção para *sprint*. A data de inserção representa o dia em que o requisito foi adicionado ao *product backlog* e a coluna data de seleção para *sprint* representa o planejamento de execução das *sprints*.

Quadro 3 – *Product Backlog*

ID	Descrição	Data da Inserção	Data de seleção para <i>Sprint</i>
1	Login	03/02/2020	03/02/2020
2	Manter Usuário	03/02/2020	03/02/2020
3	Manter Quarteirão	03/02/2020	03/02/2020
4	Manter Localidade	03/02/2020	03/02/2020
5	Manter Município	03/02/2020	03/02/2020
6	Manter Zona	03/02/2020	03/02/2020
7	Manter Imóvel	03/02/2020	10/02/2020
8	Manter Rua	03/02/2020	10/02/2020
9	Manter Atividade	03/02/2020	17/02/2020
10	Definir Ciclo	03/02/2020	17/02/2020
11	Planejar Atividade	03/02/2020	24/02/2020
12	Iniciar Trabalho Diário	03/02/2020	02/03/2020
13	Finalizar Trabalho Diário	03/02/2020	02/03/2020
14	Realizar Vistoria	03/02/2020	02/03/2020
15	Fragmentar Quarteirão	03/02/2020	09/03/2020
16	Realizar Vistoria de Laboratório	03/02/2020	09/03/2020
17	Realizar Vistoria de Campo Mobile	10/06/2020	Não definida
18	Definir Trabalho Diário	10/06/2020	Não definida
19	Visualizar Resultado da Atividade	10/06/2020	Não definida
20	Visualizar Resultado do Ciclo	10/06/2020	Não definida
21	Relatório por Ciclo	10/06/2020	Não definida
22	Manter Laboratório	10/06/2020	Não definida
23	Manter Regional de Saúde	10/06/2020	Não definida
24	Fragmentar Quarteirão	10/06/2020	Não definida
25	Enviar Amostra para Laboratório	10/06/2020	Não definida
26	Imprimir formulário de vistoria	10/06/2020	Não definida
27	Imprimir Boletim de Exames	10/06/2020	Não definida
28	Inserir Resultados das Amostras	10/06/2020	Não definida
29	Visualizar resumo do trabalho diário do município	10/06/2020	Não definida
30	Visualizar resumo do resultado de laboratório das amostras no município	10/06/2020	Não definida
31	Resumo do Trabalho Diário da equipe	10/06/2020	Não definida
32	Acompanhar demandas por laboratório	10/06/2020	Não definida
33	Visualizar Boletim de Campo	10/06/2020	Não definida
34	Relatório Diário por atividade	10/06/2020	Não definida
35	Visualizar Relatório Semanal	10/06/2020	Não definida
36	Relatório por Ciclo	10/06/2020	Não definida
37	Manter Atividade para o Município	10/06/2020	Não definida
38	Definir Perfis	10/06/2020	Não definida
39	Visualizar Resultado da Atividade no Município	10/06/2020	Não definida
40	Visualizar Resultados das Amostras	10/06/2020	Não definida
41	Registro do Trabalho Diário	10/06/2020	Não definida
42	Visualizar Resumo do Trabalho diário da Equipe	10/06/2020	Não definida
43	Visualizar resumo do resultado de laboratório de suas amostras	10/06/2020	Não definida

ID	Descrição	Data da Inserção	Data de seleção para <i>Sprint</i>
44	Acompanhar demandas por laboratório pelo supervisor	10/06/2020	Não definida
45	Relatório de Trabalho Diário	10/06/2020	Não definida
46	Registrar Dados Abertos na FIWARE	10/06/2020	Não definida
47	Resumo do Trabalho Diário	10/06/2020	Não definida
48	Realizar Exame	10/06/2020	Não definida

Fonte: os autores

Como mencionado, o *product backlog* foi realizado em seis *sprints*. O detalhamento de todas as *sprints* podem ser encontrados no repositório de arquivos apresentado neste capítulo. Cada *sprint* possui a ela associado um *sprint backlog*, que compreende um subconjunto dos requisitos do *product backlog* a ser desenvolvido na *sprint* e ao final dela, validado pelo *product owner*. A seguir são detalhadas apenas as *sprints* I e II para efeito de ilustração do que foi realizado. As demais sprints encontram-se no [Apêndice A](#).

Durante a *sprint* I, foram desenvolvidos os requisitos básicos e que são pré-requisitos para as demais etapas da ferramenta.

Quadro 4 – *Backlog* da *sprint* I

ID	Descrição	Data de inserção	Tempo Orçado	Tempo Realizado	Data de Validação	Retrabalho (sim ou não)
1	Login	03/02/2020	240 min	234 min	-	Não
2	Manter Usuário	03/02/2020	480 min	436 min	-	Não
3	Manter Quarteirão	03/02/2020	240 min	208 min	-	Não
4	Manter Localidade	03/02/2020	240 min	90 min	-	Não
5	Manter Município	03/02/2020	480 min	300 min	-	Não
6	Manter Zona	03/02/2020	240 min	90 min	-	Não

Fonte: os autores

Durante a *sprint* II,

Na *sprint* II, foi desenvolvida as demais funcionalidade básica pré-requisitos para os casos de uso críticos.

Quadro 5 – Backlog da sprint II

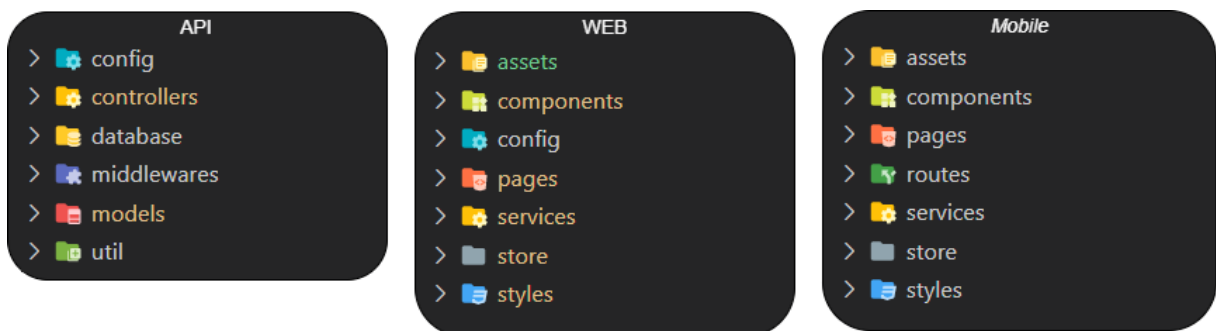
ID	Descrição	Data de inserção	Tempo Orçado	Tempo Realizado	Data de Validação	Retrabalho (sim ou não)
7	Manter Imóvel	03/02/2020	240 min	300 min	-	Não
8	Manter Rua	03/02/2020	240 min	200 min	-	Não

Fonte: os autores

3.3 Projeto da Ferramenta

O código do protótipo da ferramenta AaMT foi construído usando uma adapção do modelo MVC. As estruturas de pastas da API, WEB e *mobile* são exibidas na [Figura 11](#) para melhor explicar a adaptação MVC que foi usada no protótipo da ferramenta AaMT.

Figura 11 – Estrutura de pastas da API, aplicação WEB e *mobile*



Fonte: os autores

Na API da ferramenta não existe a estrutura de visão, já que tudo é feito através de requisições HTTP. Portanto, a pasta *controllers* desempenha o papel dos controles da arquitetura MVC, o qual solicitando mudanças de estados da aplicação ao modelo da aplicação, contido na pasta *models*. Os códigos contidos nas pastas *controllers* e *models* acessam dados ou funções nas demais pastas. A pasta *config*, contém informações de acesso a base de dados, enquanto a pasta *database*, contém os código migrações da base. As migrações são responsáveis por criar a base de dados com a estrutura definida ou realizar modificações, elas fazem parte da biblioteca *sequelize* e funcionam como um versionado da estrutura da base. Por fim, a pasta *util* desempenha o papel de armazenar funções reutilizáveis de acesso a toda a aplicação.

Já as aplicações WEB e *mobile* não contêm as estruturas de controle e modelo. Isso se deve às tecnologias *React* e *React Native* terem como objetivo proporcionar a construção da interface com o usuário em tais ambientes. Contudo, mesmo a aplicação contendo somente a estrutura de visão que interage com o usuário, faz-se necessária a comunicação com a camada de controles na API para requisição de dados ou mudança de estados da aplicação. As pastas

responsáveis pela comunicação com o controle da API são *store* e *services*. Os dados resultantes da comunicação com a API são construídos na visão que está estruturada nas pastas *components* e *pages*.

Nas próximas seções são descritas as principais funcionalidades e a estrutura lógica da ferramenta, expressa pelo diagrama de classes e diagrama de atividades e as principais funcionalidades.

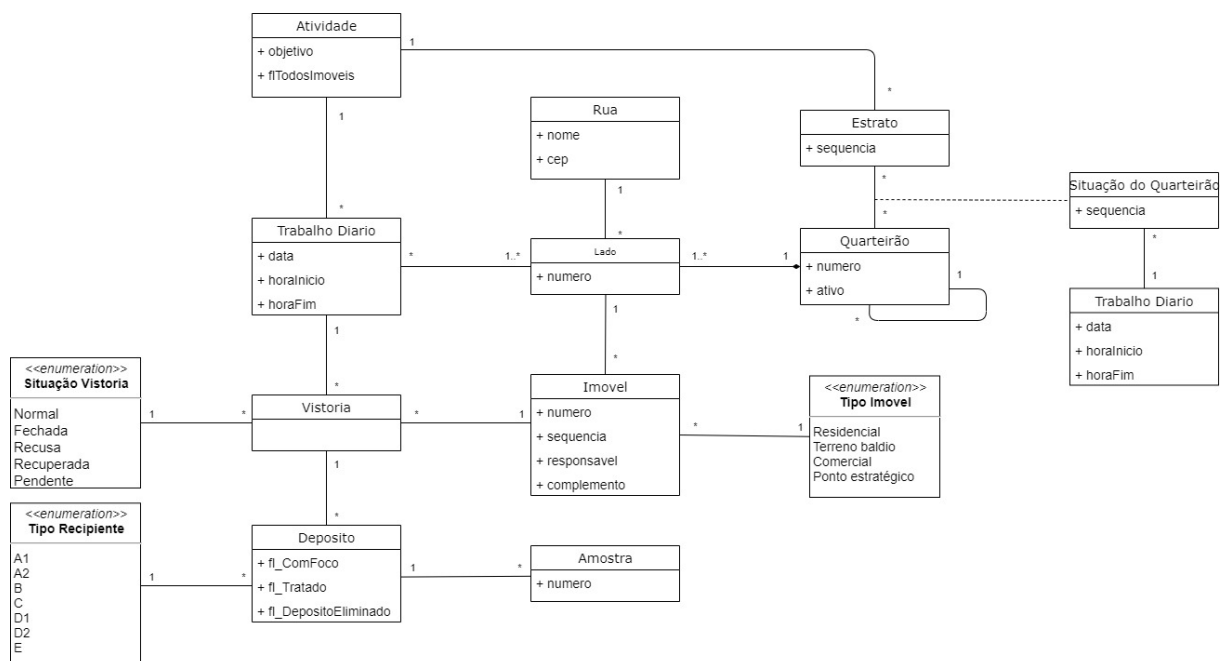
3.3.1 Diagrama de Classes

O projeto da ferramenta AaMT foi desenvolvido e refinado ao longo das seis *sprints* do projeto. O diagrama desta seção refere-se à arquitetura final da ferramenta.

Devido à complexidade do domínio do problema, a AaMT contém cinco visões de diagrama de classes divididas em: vistoria, tratamento, exame das amostras, atividades e local. Nesta seção, somente a visão de vistoria, é descrita; as demais visões encontram-se no [Apêndice A](#).

A visão de vistoria compreende as classes *quarteirão*, *lado*, *rua* e *imóvel*. Abstraindo o mundo real, diz-se que um *quarteirão* contém três ou mais *lados* e que cada *lado* está associado a uma *rua*. Ainda, um *imóvel* está localizado em *lado*, consequentemente, a um *quarteirão* e uma *rua*, conforme exibido na [Figura 12](#).

Figura 12 – Visão da vistoria do Diagrama de Classes da ferramenta AaMT



Fonte: os autores

Um trabalho diário faz relação com uma atividade a qual possui estrato. O conceito de de estrato foi projetado a partir da metodologia LIRAA, segundo o LIRAA um estrato deve respeitar o

intervalo ideal de 8000 a 12100 imóveis por estrato. Seguindo o raciocínio, o diagrama de classe projeta a relação entre estrato e bairros para divisão no município.

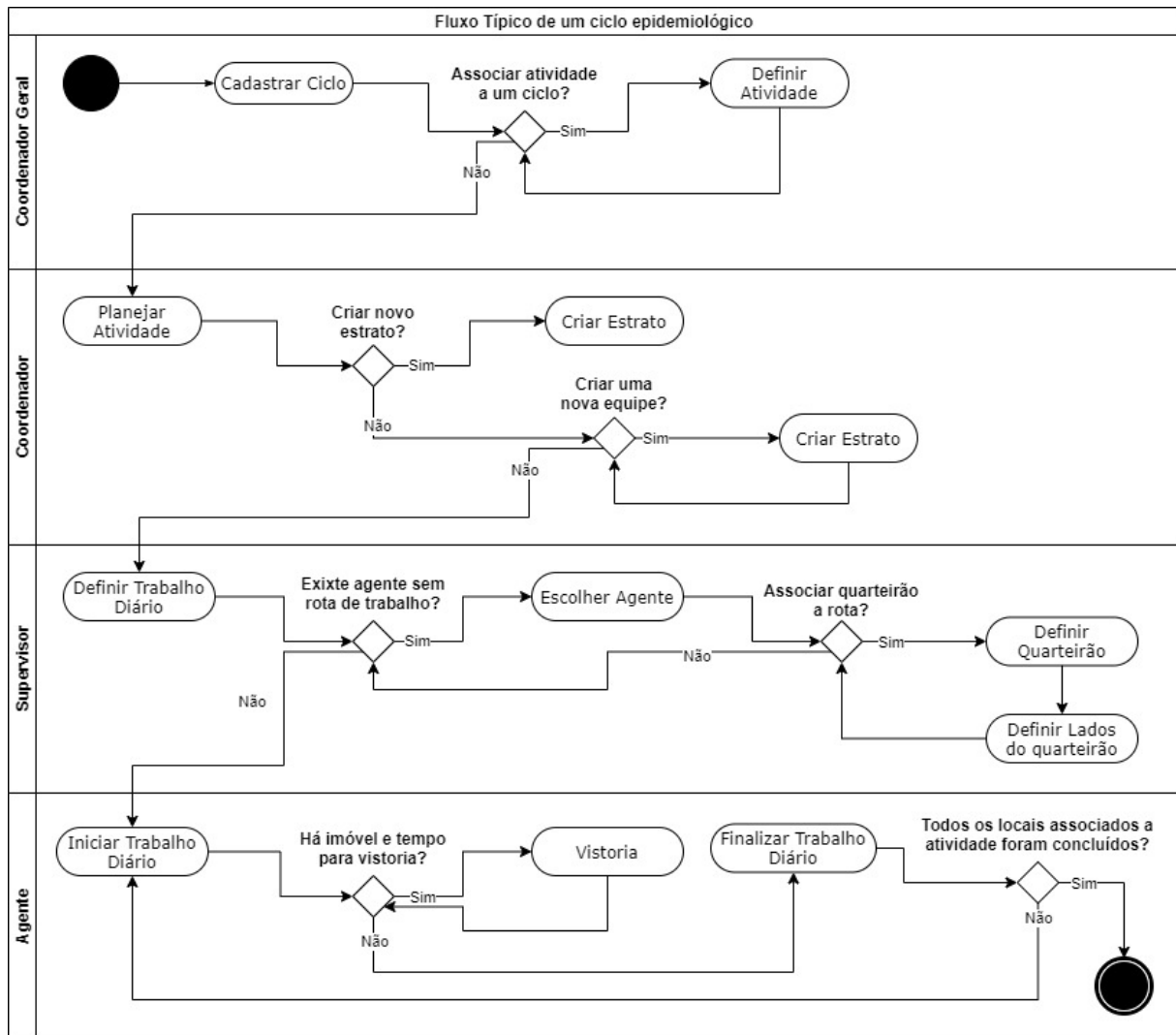
Ao iniciar um objeto da classe Trabalho Diário de um Usuário, neste caso um Agente, é necessário definir quais são os lados que serão trabalhados por este agente. Isto significa para aplicação que todos os registros de imóveis, ruas, bairros e lados, dos bairros definidos no trabalho diário, serão vistoriados pelo usuário durante a atividade de campo do dia e devem ser consultados na base e armazenados em seu dispositivo de trabalho, que pode ser um celular ou *tablet*. Essa ação é tratada no caso de uso "Iniciar trabalho diário".

Com estas informações em mãos, um agente pode sair a campo com seu aparelho *mobile* para realizar vistorias sem necessidade de Internet. Ao final do trabalho de campo, os dados coletados devem ser enviados para a API. O processo de enviar os dados para a base de dados é implementada no caso de uso "Finalizar trabalho diário". Para executar esta ação, o dispositivo *mobile* deve estar conectado a internet. O aplicativo então envia as vistorias realizadas no dia para a API e atualiza o campo "hora fim" com a hora atual na classe "Trabalho Diário", conforme a [Figura 12](#). Após a finalização do trabalho diário, os imóveis cujas as vistorias não foram executadas - situação igual a pendente - deverão ser incluídos em outros trabalhos diários. Tal informação pode ser visualizada no "Relatório de trabalho diário".

3.3.2 Diagrama de Atividades

Para compreensão das atividades executadas pela AaMT foi adotado um fluxo típico, exibido na [Figura 13](#). Este diagrama possibilita o entendimento geral do cadastro de ciclo e os passos até a realização das vistorias.

Figura 13 – Fluxo típico da ferramenta



Fonte: os autores

O fluxo é dividido em quatro partições que são: Coordenador Geral, Coordenador, Supervisor e Agentes, observe a [Figura 13](#). O fluxo se inicia com o cadastro de ciclo e suas atividades. Nesta etapa, o responsável pelo cadastro é o coordenador geral que pertence a uma regional de saúde. As atividades, por ele definidas neste ciclo, serão aplicadas a todos os municípios vinculados à regional de saúde.

A segunda partição, Coordenador, é iniciada pela atividade de planejar atividade. Durante este processo o coordenador deve criar pelo menos um estrato. Um estrato contém vários locais e sua criação obriga que o coordenador associe todos os locais do estrato a uma equipe que irá executar a vistoria em cada local.

Com o ciclo e a atividade registrada o supervisor é responsável por realizar o planejamento das rotas diárias dos agentes, definido pela atividade 'Definir Trabalho Diário'. Durante este processo o supervisor deve associar os quarteirões e os lados que um agente irá trabalhar durante o dia.

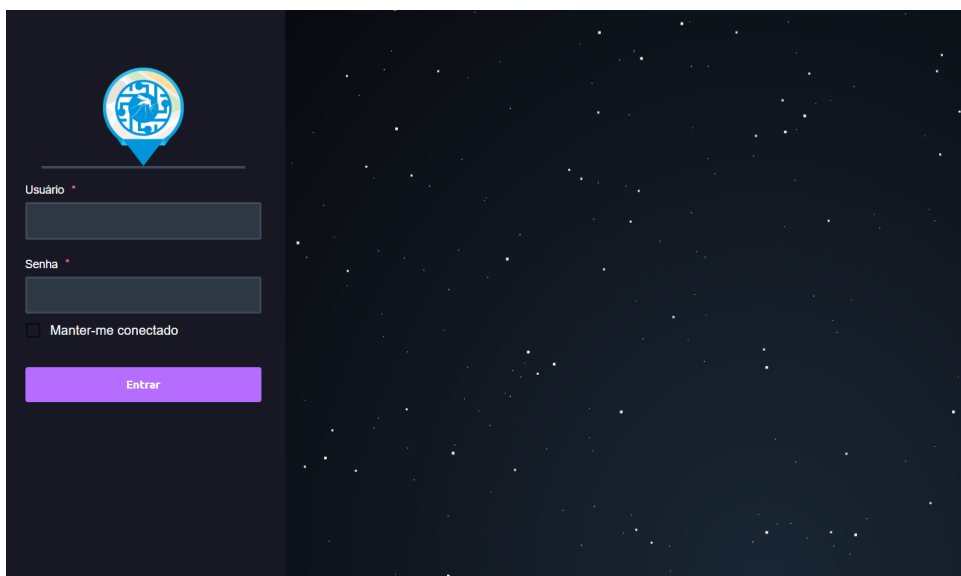
Por fim, a partição Agente é composta pelo componente de vistoria de campo. Esta partição se inicia com a atividade iniciar trabalho diário de um agente, esta ação registra o horário do início de trabalho de campo e realiza o *download* de todos os registro de quarteirões, lados e imóveis associado ao trabalho diário do agente. Após este processo o agente vai a campo realizar as vistorias. No final do dia o agente, em posse de internet, executa a atividade finalizar trabalho do dia, onde são sincronizadas todas as vistorias com o servidor.

3.4 Principais funcionalidades

O projeto AaMT possui um conjunto de funcionalidades necessárias para a solução do problema aqui exposto. Nesta seção, são apresentadas as principais funcionalidades que já foram implementadas.

A ferramenta deve possuir um controle de autenticação de usuário, como mostra a [Figura 14](#), isto serve para garantir a autorização de acesso ao sistema, bem como a definição do nível de acesso. Ao realizar a autenticação o usuário recebe um *token*¹ que é armazenado para que seja permitido a realização de requisições futuras à API.

Figura 14 – Tela de login dos usuários



Fonte: os autores

A [Figura 13](#) descreve desde a criação de um ciclo até a realização das vistorias. O cadastro de ciclo necessita que o usuário esteja logado e contenha o perfil de coordenador geral. Na definição do ciclo, o coordenador geral deve definir as atividades que ele julga necessário serem executadas em todos os municípios de sua responsabilidade, como é possível visualizar na [Figura 15](#).

¹ Token: chave de acesso a aplicação.

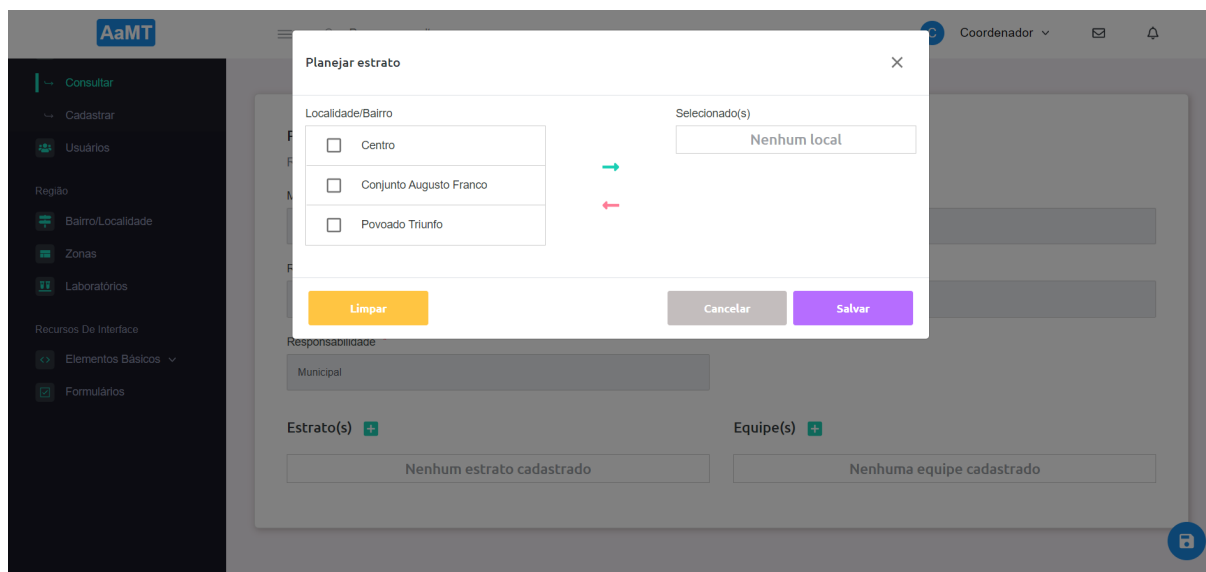
Figura 15 – Tela de cadastro de um novo ciclo

Fonte: os autores

É possível registrar a execução de mais de uma atividade por ciclo, basta clicar no botão ao lado do título atividades, como mostra a [Figura 15](#). A quantidade de atividades e quais atividades serão executadas é de responsabilidade do coordenador geral; a ferramenta deixa livre a escolha delas.

O coordenador municipal pode visualizar as atividades em aberto para o município e realizar o planejamento delas. Uma atividade pode conter três abrangências: por localidade, por zona ou por quarteirão. A abrangência da atividade afetará em quais os locais irão aparecer para associação no planejamento, por exemplo: ao escolher a abrangência por localidade, durante a criação de estrato, de um planejamento de atividade, será exibido a lista de localidades/bairros do município. Este exemplo é mostrado na [Figura 16](#) onde é possível ver os bairros de Simão Dias, pois a atividade em questão foi definida por localidade.

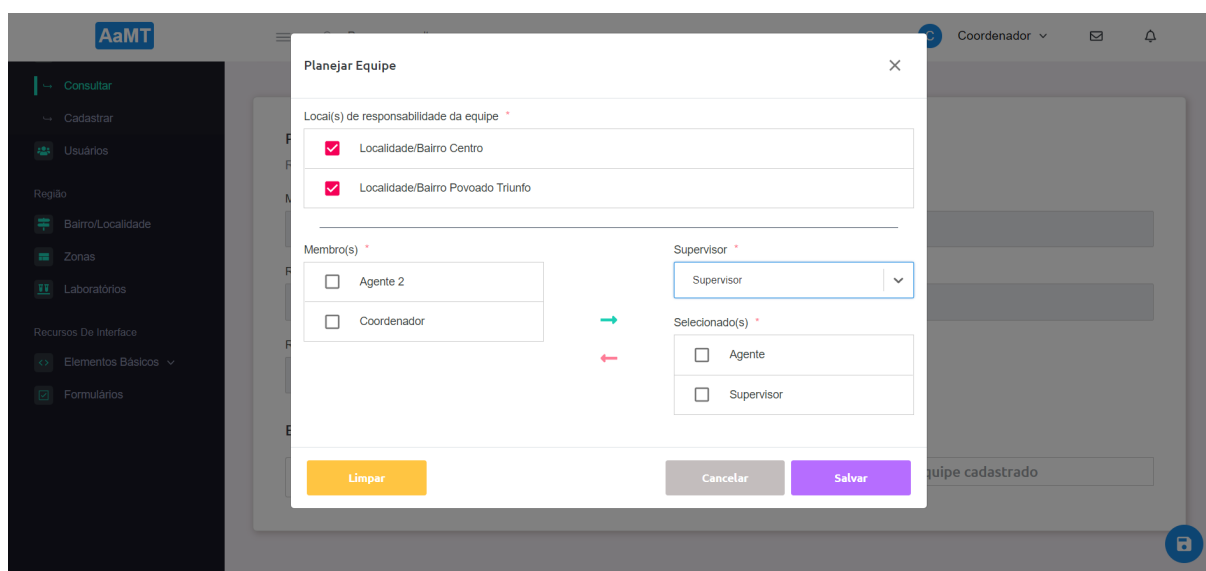
Figura 16 – Tela de planejamento de um estrato



Fonte: os autores

Uma atividade pode conter mais de um estrato, um estrato representa um conjunto de locais selecionados para executar a atividade. A Cada local selecionado no estrato, [Figura 16](#), deve ser associado uma equipe.

Figura 17 – Tela de planejamento de equipes



Fonte: os autores

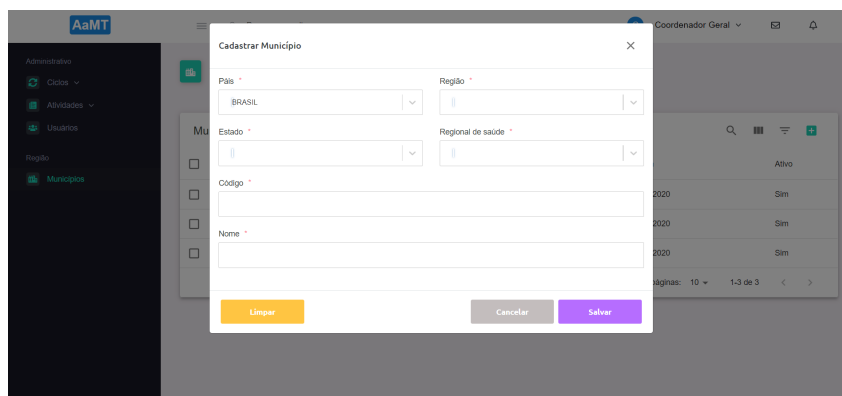
A [Figura 17](#) apresenta o registro de uma equipe responsável por dois bairros de Simão Dias e é composta por dois usuário, um supervisor e um agente. O planejamento de uma atividade finaliza quando não há mais nenhum local, pertencente a um estrato, que não está na responsabilidade de uma equipe.

Para a execução da atividade, os supervisores devem definir as rotas que os agentes irão trabalhar. Durante o processo o supervisor escolhe qual atividade do município, que está em aberto, ele irá trabalhar no dia, em seguida escolhe a equipe, o agente, os bairros e os lados e este agente irá trabalhar. O processo de definir trabalho diário se repete por todos os agentes que irão a campo no dia. Vale ressaltar que os bairros exibidos serão somente os de responsabilidade da equipe a qual o supervisor está associado.

Um agente com sua rota de trabalho do dia definido solicita o início do trabalho de campo, ao fazer isso a ferramenta deve realizar o *download* das informações contidas na base necessárias para uma vistoria como: bairros e seus lados e os imóveis que foram associados ao trabalho diário. Os agentes, então, podem ir a campo realizar as vistorias nos imóveis. Ao final do dia, o agente deve estar com acesso a internet para realizar o processo de finalizar trabalho. Ao iniciar esta etapa, a ferramenta faz envio de todos os dados coletados em campo e sincroniza com a base de dados através da API. Este processo se repete até que todos os bairros da atividade estejam em situação de concluído.

Outra funcionalidade da ferramenta é o cadastro de município, realizado pelo coordenador geral. Como pode ser observado na [Figura 18](#), um município pode conter mais de um coordenador municipal, assim como a regional de saúde, basta que o coordenador geral cadastre mais de um usuário com perfil de coordenador no mesmo município.

Figura 18 – Tela de cadastro de município



Fonte: os autores

A ferramenta também dispõe o cadastro de localidade/bairro, [Figura 19](#). Uma localidade pode ser categorizada em rural, terra indígena ou urbana.

Figura 19 – Tela de cadastro de localidade

A interface de usuário da ferramenta AaMT apresenta uma barra lateral esquerda com o menu de navegação. O formulário principal, 'Cadastrar Localidade', contém os seguintes elementos:

- Município:** Simão Dias
- Código da localidade/bairro:** Campo de texto vazio.
- Categoria da localidade:** Menu suspenso com uma seta para baixo.
- Nome:** Campo de texto vazio.
- Botões:** 'Ligar' (laranja), 'Cancelar' (cinza) e 'Salvar' (roxo).

No plano de fundo, é visível uma tabela de dados:

Município	Ativo
Simão Dias	Sim
Simão Dias	Sim
Simão Dias	Sim

Na base da interface, há uma barra de status que indica 'Linhas por página: 10' e '1-3 de 3'.

Fonte: os autores

A ferramenta deve conter o cadastro de quarteirões e imóveis realizados pelo supervisor, observe a [Figura 20](#). Esses cadastros envolvem, também, o cadastro de lados e ruas necessários na composição do quarteirão. Durante o processo de planejamento, foi discutido como seria possível que dois ou mais agentes fossem a campo no mesmo quarteirão sem que houvesse duplicata de vistoria no mesmo imóvel, já que a vistoria ocorre de forma *offline*. A solução foi projetar atividade de definição de rotas, definir trabalho diário, esta atividade foi vista durante a visita técnica, ocorrida na produção desta monografia, onde os supervisores definiram onde os agentes trabalhariam.

Mesmo após a solução de trabalho diário, nos deparamos na situação de como dois agentes trabalhariam em dois quarteirões ao mesmo tempo. Desta forma, optamos por expandir o conceito de quarteirão e agregar lados e ruas, tornando o imóvel pertencente a um lado, não diretamente a um quarteirão, tornando assim possível dois agentes trabalhar em mesmo quarteirão, desde que seja em lados direferente.

Figura 20 – Tela de cadastro de quarteirão

AaMT

☰

🔍 Procurar consulta

S

Supervisor

✕

✉

🔔

Imóveis

+

Informações Do Quarteirão

Localidade

▼

Zona

▼

Número

Lados

Rua

▼

CEP

Outra

Número do lado

+

Fonte: os autores

4

Conclusão

Para auxiliar o combate a endemias provocadas pelo *Aedes aegypti* esta proposta de TCC se propõe a projetar uma ferramenta que automatize as tarefas dos ACEs e disponibilize dados abertos para subsidiar atividades realizadas pelo grupo de pesquisa liderado pela professora Dra. Roseli Lacorte dos Santos. O sistema proposto visa cadastrar as tarefas, vinculando equipes para realizá-las. As tarefas são baseadas no formulário do LIRAA e PNCD, ver [Anexo A](#) e [Anexo B](#). Ou seja, os coordenadores municipais dos ACEs definirão a metodologia da atividade que será executada conforme a necessidade do município. Outra atribuição do software, é o cálculo de índices e mapeamento dos focos, realizados pelos agentes.

Para a consecução deste trabalho, foi feito um estudo bibliográfico das metodologias para o combate ao vetor *Aedes aegypti* (LIRAA e PNCD) vigentes nos centros de combate a endemias. Essas etapas serviram para elucidar os requisitos do software e as funcionalidades (*backlog*) do produto.

Durante o planejamento da ferramenta ocorreram reuniões para dúvidas e correções com a Dra. Roseli Santo do DMO, Sidney Sa e José Santos da Secretaria de Saúde do estado de Sergipe. Devido a atrasos no planejamento da ferramenta e a pouca disponibilidade do funcionários da Secretaria de Saúde não houve tempo hábil para a validação do protótipo.

O protótipo desenvolvido contém partes da ferramenta, porém ainda é necessário validar e desenvolver as demais funcionalidade projetadas nesta monografia, que são: o componente de relatório da ferramenta, cálculo dos índices e finalizar os componentes de vistorias. O protótipo da AaMT implementada os modulos de gestão e controle dos ciclos epidemiológicos e atividades que são pré-requisitos para as funcionalidades de vistoria, registro de exames, relatórios e cálculo dos índices.

É necessário realizar o desenvolvimento e validação da ferramenta com dados reais para que, juntamente com a professora Dra. Roseli Santo, seja implantado nos municípios de Sergipe, iniciando primeiramente em Simões Dias. A ferramenta proposta pretende tornar possível a

automação das atividades realizadas pelos ACEs, influenciando positivamente no combate ao mosquito *Aedes aegypti*.

Além disso, como trabalhos futuros, existem outras atividades realizadas no LEPaT, coordenadas pela professora Dra. Roseli Santos, como a instalação de ovitrampa nos imóveis. Esta atividade instala ovitrampas em regiões para que as fêmeas depositem seus ovos e os alunos ou ACEs possam recolher, os mesmos, e mapear os locais que possuem fêmeas. Segundo Dra. Roseli Santos existem outras atividades que não foram mapeadas para a ferramenta e podem servir como base para execução de outros projetos do DMO.

Seria possível reliazar a implantação destas atividades fazendo o mapeamento de quais componentes elas possuem de diferente das metodologias LIRAA e PNCD. Por exemplo, todos possuem o componente de imóvel, este componente condiz em registrar qual o imóvel da vistoria, já a atividade de instalação de ovitrampa, possui formulário durante a instalação e remoção. Este processo deve ser mapeado para o projeto da ferramenta.

Uma ideia similar a esta poderia ser utilizada para realizar a implantação de uma nova metodologia, assim como o LIRAA e o PNCD, para registrar as pessoas que possuem suspeita ou confirmação de corona vírus ou outras doenças como tuberculose, influenza e outras doenças contagiosas.

Referências

CKAN. *About CKAN*. 2019. Disponível em: <<https://ckan.org/about/>>. Acesso em: 19 ago. 2019. Citado na página 32.

CODECO, C. et al. "infodengue: A nowcasting system for the surveillance of arboviruses in brazil". *Revue d'Épidémiologie et de Santé Publique*, v. 66, p. S386, 2018. ISSN 0398-7620. European Congress of Epidemiology "Crises, epidemiological transitions and the role of epidemiologists". Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0398762018311088>>. Citado na página 39.

DIALYNAS, E. et al. Miro and irbase: It tools for the epidemiological monitoring of insecticide resistance in mosquito disease vectors. *PLoS neglected tropical diseases*, v. 3, p. e465, 02 2009. Citado na página 40.

EIRAS; RESENDE. *Preliminary evaluation of the 'Dengue-MI' technology for Aedes aegypti monitoring and control*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Departamento de Parasitologia, Instituto de Ciências Biológicas, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil, 2009. Citado 2 vezes nas páginas 15 e 39.

FACEBOOK. *React - Uma biblioteca JavaScript para criar interfaces de usuário*. 2019. Disponível em: <<https://pt-br.reactjs.org/>>. Acesso em: 21 ago. 2019. Citado na página 30.

FACEBOOK. *React Native - Learn once, write anywhere*. 2019. Disponível em: <<https://facebook.github.io/react-native/>>. Acesso em: 21 ago. 2019. Citado na página 30.

FIWARE. *Tour Guid*. 2019. Disponível em: <<https://www.fiware.org/developers/tour-guide/>>. Acesso em: 19 ago. 2019. Citado 2 vezes nas páginas 16 e 32.

FOWLER, M. *UML Essencial*. [S.l.]: Porto Alegre, RS, BRA: Bookman, 2005. v. 3 ed. Citado na página 38.

FOWLER, M. *UML Essencial 3ed*. Porto Alegre, RS, BRA: Bookman, 2005. Citado na página 34.

GALÁN, F. et al. Exploiting the fiware cloud platform to develop a remote patient monitoring system. In: . [S.l.: s.n.], 2015. Citado na página 40.

HAMER, S.; CURTIS-ROBLES, R.; HAMER, G. Contributions of citizen scientists to arthropod vector data in the age of digital epidemiology. *Current Opinion in Insect Science*, v. 28, 05 2018. Citado na página 40.

IBM. *Diagramas de Classes*. 2016. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS8PJ7_9.6.0/com.ibm.xtools.modeler.doc/topics/cclassd.html>. Acesso em: 11 abr. 2020. Citado na página 36.

INSOMNIA. *Insomnia*. 2019. Disponível em: <<https://support.insomnia.rest/>>. Acesso em: 21 ago. 2019. Citado na página 31.

JACYNTHO, M. D.; SCHWABE, D.; ROSSI, G. A software architecture for structuring complex web applications. *J. Web Eng.*, v. 1, p. 37–60, 01 2002. Citado 2 vezes nas páginas 29 e 30.

LIRAA. BRASIL. Ministério da Saúde. *Levantamento Rápido de Índices para Aedes aegypti - LIRAA - Para Vigilância Entomológica do Aedes aegypti no Brasil. Metodologia para Avaliação dos Índices de Breteau e Predial e Tipo de Recipientes. 1ª ed. MS. Brasília - DF, 2013. 2013. Citado 3 vezes nas páginas 15, 21 e 23.*

MICROSOFT. *Visual Studio Code*. 2019. Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 21 ago. 2019. Citado na página 31.

MS. *Orientações técnica para utilização do larvicida pyriproxyfen (0,5 G) no controle de Aedes aegypti*. 2013. Disponível em: <<https://www.saude.gov.br/saude-de-a-z/gripe/926-saude-de-a-z/controle-de-vetores-inseticidas-e-larvicidas/>>. Acesso em: 21 mai. 2020. Citado na página 40.

MS. BRASIL. Ministério da Saúde. *Boletins epidemiológicos. BE Vol. 50 Nº 13, 2019: Monitoramento dos casos de arboviroses urbanas transmitidas pelo Aedes (dengue, chikungunya e Zika) até a Semana Epidemiológica 12 de 2019*. 2019. Citado 3 vezes nas páginas 14, 15 e 25.

MS. BRASIL. Ministério da Saúde. *Combate ao Aedes aegypti: prevenção e controle da Dengue, Chikungunya e Zika*. 2019. Disponível em: <<http://www.saude.gov.br/saude-de-a-z/aedes-aegypti>>. Acesso em: 7 ago. 2019. Citado 2 vezes nas páginas 19 e 20.

NODEJS. *NodeJS*. 2019. Disponível em: <<https://nodejs.org/en/about/>>. Acesso em: 21 ago. 2019. Citado na página 30.

PAGOTTO, T. et al. Scrum solo: Software process for individual development. In: . [S.l.: s.n.], 2016. p. 1–6. Citado 2 vezes nas páginas 27 e 29.

PLAMANESCU, R. et al. A synchronized measurements fiware platform for smart grid applications. In: . [S.l.: s.n.], 2019. p. 1–5. Citado na página 34.

PNCD. BRASIL. Ministério da Saúde; Brasil. *Fundação Nacional de Saúde - FUNASA. Programa Nacional de Controle da Dengue. Vigilância Epidemiológica [s.n.]. FUNASA, 2002. 2002. Citado 2 vezes nas páginas 15 e 24.*

RUIZ, M. A. et al. Environmental monitoring based on fiware: A medical case study. In: . [S.l.: s.n.], 2018. p. 1–6. Citado na página 40.

SAGE. *SAGE - Sala de Apoio à Gestão Estratégica. Índice de Desempenho do Sistema Único de Saúde*. 2011. Disponível em: <i3geo.saude.gov.br/i3geo/sage/abremapa.php?id=1>. Acesso em: 30 jul. 2019. Citado na página 14.

SEQUELIZE. *Sequelize*. 2020. Disponível em: <<https://sequelize.org/v5/>>. Acesso em: 09 abr. 2020. Citado na página 32.

SRIVASTAVA, A.; BHARDWAJ, S.; SARASWAT, S. Scrum model for agile methodology. In: . [S.l.: s.n.], 2017. p. 864–869. Citado 2 vezes nas páginas 26 e 27.

STRONGLOOP, I. *Express*. 2020. Disponível em: <<https://expressjs.com/pt-br/>>. Acesso em: 09 abr. 2020. Citado na página 32.

VALERO, N. R. et al. Mobile based surveillance platform for detecting zika virus among spanish delegates attending the rio de janeiro olympic games. *PLOS ONE*, v. 13, p. e0201943, 08 2018. Citado na página 40.

Apêndices

APÊNDICE A – Documentação complementar da ferramenta AaMT

Este documento apresenta todos os documentos gerados durante o planejamento da ferramenta AaMT.

A.1 Sprints

Durante a execução do projeto foram realizadas seis *sprints*. As *sprints* são a etapa posterior a atividade *requirements*, de acordo a Scrum Solo. Durante a *sprint* I, foram desenvolvidos os requisitos básicos e que são pré-requisitos para as demais etapas da ferramenta.

Figura 21 – *Backlog* da *sprint* I

ID	Link para planta de especificação	Data de inserção	Tempo de construção (em minutos)		Data de Validação	Retrabalho (sim ou não)
			Orçado	Realizado		
1		03/02/2020	240	234	10/02/2020	não
2		03/02/2020	480	436	10/02/2020	não
3		03/02/2020	240	208	10/02/2020	não
4		03/02/2020	240	90	10/02/2020	não
5		03/02/2020	480	300	10/02/2020	não
6		03/02/2020	240	90	10/02/2020	não

Fonte: os autores

Na *sprint* II, após finalizar o desenvolvimento dos requisitos básicos da ferramenta possibilitando, assim, iniciar o desenvolvimento críticos para o funcionamento da ferramenta.

Figura 22 – Backlog da *sprint* II

ID	Link para planta de especificação	Data de inserção	Tempo de construção (em minutos)		Data de Validação	Retrabalho (sim ou não)
			Orçado	Realizado		
7		03/02/2020	240	300	17/02/2020	não
8		03/02/2020	480	960	17/02/2020	não

Fonte: os autores

Durante a *sprint* III, foi desenvolvido dois requisitos críticos da ferramenta AaMT, a definição de ciclo e gestão das atividades. Estes requisitos compreendem grande parte das funções de um coordenador geral.

Figura 23 – Backlog da *sprint* III

ID	Link para planta de especificação	Data de inserção	Tempo de construção (em minutos)		Data de Validação	Retrabalho (sim ou não)
			Orçado	Realizado		
10		03/02/2020	480	400	24/02/2020	não
9		03/02/2020	960	1000	24/02/2020	não

Fonte: os autores

Durante a *sprint* IV, foi desenvolvido o requisito de planejamento de atividades, onde foi necessário trabalhar com a criação de estratos e equipes. A execução desta *sprint* exigiu requisitos como manter atividade, manter quartirão, manter localidade e manter zona.

Figura 24 – Backlog da *sprint* IV

ID	Link para planta de especificação	Data de inserção	Tempo de construção (em minutos)		Data de Validação	Retrabalho (sim ou não)
			Orçado	Realizado		
11		03/02/2020	1200	1090	02/03/2020	não

Fonte: os autores

A *sprint* V, compreendeu todo o módulo de vistoria de campo da ferramenta, um requisito crítico. Foi necessário a criação da aplicação *mobile* durante a execução da *sprint*.

Figura 25 – Backlog da *sprint* V

ID	Link para planta de especificação	Data de inserção	Tempo de construção (em minutos)		Data de Validação	Retrabalho (sim ou não)
			Orçado	Realizado		
12		03/02/2020	480	500	09/03/2020	não
14		03/02/2020	1200	1100	09/03/2020	não
13		03/02/2020	480	300	09/03/2020	não

Fonte: os autores

A *sprint* VI, possui em seu *sprint backlog* requisitos que necessitavam dos demais prontos antes da sua criação, como a realização dos exames, bem como requisitos de menor prioridade para o *product owner*, como a fragmentação de quarteirão¹.

Figura 26 – Backlog da *sprint* VI

ID	Link para planta de especificação	Data de inserção	Tempo de construção (em minutos)		Data de Validação	Retrabalho (sim ou não)
			Orçado	Realizado		
12		03/02/2020	480	500	09/03/2020	não
14		03/02/2020	1200	1100	09/03/2020	não
13		03/02/2020	480	300	09/03/2020	não

Fonte: os autores

A.2 Casos de Uso

Nesta seção, é descrito todos os casos de uso apresentados na [Figura 10](#).

A.2.1 Manter Município

Um município pertence a uma regional de saúde. A responsabilidade de manter os municípios é dos coordenadores gerais. Este caso de uso é descrito como segue:

- **Pré-condições:** o coordenador geral está logado e com *token* de validação armazenado;

¹ Fragmentar um quarteirão: processo de dividir um quarteirão em dois ou unir dois ou mais quarteirões em um. Isto ocorre quando ocorre de criar uma rua entre o quarteirão ou quando a rua que separava quarteirões é fechada.

- **Pós-condições:** os dados do município foram armazenados.
- **Cenário principal:**
 1. O coordenador geral acessa a tela de cadastro de município;
 2. O sistema exibe a tela de cadastro de município;
 3. O coordenador geral preenche o formulário;
 4. O sistema salva as informações e retorna mensagem de sucesso ao coordenador geral.

A.2.2 Manter Usuário

Os usuários são mantidos por coordenador geral ou municipal. O coordenadores gerais têm acesso irrestrito ao cadastro de usuários e os coordenadores municipais só é permitido cadastrar supervisores ou agentes, no mesmo município.

- **Pré-condições:** o coordenador geral ou municipal está logado e com *token* de validação armazenado;
- **Pós-condições:** os dados do usuário foram armazenados.
- **Cenário principal:**
 1. O coordenador acessa a tela de cadastro de usuário;
 2. O sistema exibe tela de cadastro de usuário;
 3. O coordenador preenche o formulário;
 4. O sistema salva as informações e retorna mensagem de sucesso ao coordenador.

A.2.3 Manter Localidade

Um localidade/bairro contém bairros. Uma localidade pode ser categorizada como rural, urbana ou zona indígena.

- **Pré-condições:** o coordenador municipal está logado e com *token* de validação armazenado;
- **Pós-condições:** os dados da localidade foram armazenados.
- **Cenário principal:**
 1. O coordenador acessa a página de cadastro de localidade;
 2. O sistema exibe tela de cadastro de localidade;
 3. O coordenador preenche o formulário;
 4. O sistema salva as informações e retorna mensagem de sucesso ao coordenador.

A.2.4 Manter Zona

As zonas são um conjunto de quarteirões, criadas pelo os coordenadores, usadas para refinar os relatórios.

- **Pré-condições:** o coordenador está logado e com *token* de validação armazenado;
- **Pós-condições:** os dados da zona foram armazenados.
- **Cenário principal:**
 1. O coordenador clica no botão de cadastro de zona;
 2. O sistema exibe tela de cadastro de zona;
 3. O coordenador confirma a criação de uma nova zona;
 4. O sistema verifica o próximo nome de zona disponível no sistema e armazena a nova zona;
 5. O sistema retorna mensagem de sucesso para o coordenador.

A.2.5 Manter Laboratório

O coordenador é responsável por registrar os laboratórios que o município irá enviar as amostras para realização dos exames.

- **Pré-condições:** o coordenador está logado e com o *token* de validação armazenado;
- **Pós-condições:** os dados do laboratório foram armazenados.
- **Cenário principal:**
 1. O coordenador acessa a página de cadastro de laboratório;
 2. O sistema exibe tela de cadastro de laboratório;
 3. O coordenador preenche o formulário;
 4. O sistema salva as informações e retorna mensagem de sucesso ao coordenador.

A.2.6 Manter Quarteirão

É de responsabilidade do supervisor manter os quarteirões. Os quarteirões pertencem a uma localidade e possuem mais de um lado.

- **Pré-condições:** o supervisor está logado e com *token* de validação armazenado.
- **Pós-condições:** os dados do quarteirão foram armazenados.

- **Cenário principal:**

1. O supervisor acessa a página de cadastro de quarteirão;
2. O sistema exibe a tela de cadastro de quarteirão
3. O supervisor informa o CEP de uma rua;
4. O sistema consulta o CEP e retorna o nome da rua;
5. O supervisor cadastrar o lado do quarteirão e preenche demais campos do formulário;
6. O sistema salva as informações e retorna mensagem de sucesso ao supervisor.

- **Fluxos Alternativos:**

- (4) O supervisor retorna para o passo 2.

A.2.7 Manter Rua

O supervisor deve manter as ruas do município. Uma rua pertence a uma localidade e pode ser associada a mais de um lado de quarteirões.

- **Pré-condições:** o supervisor está logado e com o *token* de validação armazenado.

- **Pós-condições:** os dados da rua foram armazenadas.

- **Cenário principal:**

1. O supervisor acessa a página de cadastro da rua;
2. O sistema exibe a tela de cadastro de rua;
3. O supervisor informa o CEP;
4. O sistema consulta o CEP e retorna o nome da rua;
5. O supervisor confirma o nome da rua;
6. O sistema salva as informações e retorna mensagem de sucesso ao supervisor.

A.2.8 Manter Imóvel

Durante a primeira execução do sistema é preferível que os supervisores vão a campo cadastrando todos os imóveis do município.

- **Pré-condições:** o supervisor está logado e com o *token* de validação armazenado;

- **Pós-condições:** os dados do imóvel foram armazenados.

- **Cenário principal:**

1. O supervisor acessa a página de cadastro de imóvel;
2. O sistema exibe a tela de cadastro de imóvel;
3. O supervisor preenche o formulário;
4. O sistema salva as informações e retorna mensagem de sucesso ao supervisor.

A.2.9 Visualizar Resultado da Atividade

O sistema deve provê gráficos para apresentar os resultados das atividades que o coordenador geral solicitou aos municípios.

- **Pré-condições:** o coordenador geral está logado e com o *token* de validação armazenado;
- **Pós-condições:** os índices e relatórios são calculados e apresentados.
- **Cenário principal:**
 1. O usuário acessa a página de resultados das atividades;
 2. O sistema exibe tela com todas as atividades;
 3. O usuário escolhe a atividade que deseja visualizar o resultado;
 4. O sistema calcula e apresenta os resultados coletados durante a execução da atividade.

A.2.10 Visualizar Resultado do Ciclo

O coordenador pode visualizar os índices e relatórios de um ciclo finalizado.

- **Pré-condições:** o coordenador geral está logado e com o *token* de validação armazenado;
- **Pós-condições:** os dados são consultados por ciclo, calculados e apresentados.
- **Cenário principal:**
 1. O coordenador geral acessa a página de resultado por ciclo;
 2. O sistema exibe a tela com todos os ciclos finalizados do regional de saúde;
 3. O coordenador geral escolhe o ciclo que deseja visualizar;
 4. O sistema consulta, calcula e apresenta os resultados coletados durante o ciclo.

A.2.11 Manter Regional de Saúde

O coordenador geral tem a responsabilidade de manter o seu e demais regionais de saúde.

- **Pré-condições:** o coordenador geral está logado e com o *token* de validação armazenado;

- **Pós-condições:** os dados da regional de saúde foram armazenados.
- **Cenário principal:**
 1. O coordenador geral acessa a página de cadastro de regional de saúde;
 2. O sistema exibe a tela de cadastro de regional de saúde;
 3. O coordenador geral preenche o formulário;
 4. O sistema salva as informações e retorna mensagem de sucesso.

A.2.12 Relatório por Ciclo

O coordenador geral ou municipal pode visualizar os relatórios por ciclos em aberto ou finalizados.

- **Pré-condições:** o coordenador geral ou municipal está logado e com o *token* de validação armazenado;
- **Pós-condições:** os dados são consultados, calculados e apresentados.
- **Cenário principal:**
 1. O usuário acessa a página de relatório por ciclo;
 2. O sistema exibe tela com os ciclos;
 3. O usuário escolhe o ciclo que deseja ver o relatório;
 4. O sistema consulta os dados, calcula os índices, gera os relatórios e apresenta a tela de resultados;

A.2.13 Manter Atividade para Município

O coordenador municipal pode cadastrar atividades para seu município apenas.

- **Pré-condições:** o coordenador municipal está logado e com o *token* de validação armazenado;
- **Pós-condições:** os dados da atividade foram armazenados.
- **Cenário principal:**
 1. O usuário acessa a página de cadastro de atividade;
 2. O sistema exibe a tela de cadastro de atividade;
 3. O usuário preenche o formulário;
 4. O sistema salva as informações e retorna mensagem de sucesso.

A.2.14 Definir Perfis

O coordenador geral ou municipal tem a permissão de cadastrar ou alterar usuários com perfis iguais ou com menos permissões.

- **Pré-condições:** o coordenador geral ou municipal está logado e com o *token* de validação armazenado;
- **Pós-condições:** os dados do usuário foram armazenados.
- **Cenário principal:**
 1. O usuário acessa a página de edição de usuário;
 2. O sistema exibe a tela de edição de usuário;
 3. O usuário altera o perfil do usuário;
 4. O sistema salva as informações e retorna mensagem de sucesso.

A.2.15 Visualizar resumo do trabalho diário do município

Após as atividades de campo do dia o coordenador municipal pode solicitar a geração de relatório diário.

- **Pré-condições:** o coordenador municipal está logado e com o *token* de validação armazenado;
- **Pós-condições:** relatório diário do município gerado.
- **Cenário principal:**
 1. O usuário solicita a geração do relatório diário;
 2. O sistema consulta todos os dados gerados durante o dia informado pelo usuário e gera o relatório diário,

A.2.16 Visualizar Resumo do trabalho diário do município

Sistema deve provê apresentação de gráficos e índices resumidos de um trabalho diário.

- **Pré-condições:** o coordenador municipal está logado e com o *token* de validação armazenado;
- **Pós-condições:** resumo gerado e apresentado.
- **Cenário principal:**

1. O usuário solicita o resumo do trabalho diário do município;
2. O sistema consulta os dados, gera um resumo de todos os dados coletados e apresenta a tela de resumo do trabalho diário;

A.2.17 Visualizar Resultado da Atividade no Município

O coordenador municipal pode visualizar os resultados de uma atividade finalizada ou não concluída.

- **Pré-condições:** o coordenador municipal está logado e com o *token* de validação armazenado;
- **Pós-condições:** resultados da atividade gerados e apresentados.
- **Cenário principal:**
 1. O usuário acessa a página de atividades;
 2. O sistema exibe a tela de atividades;
 3. O usuário escolhe uma atividade finalizada ou não concluída;
 4. O sistema consulta os resultados da atividade e apresenta para o usuário.

A.2.18 Acompanhar demandas por Laboratório

O coordenador municipal podem acompanhar as demandas de amostras por laboratório.

- **Pré-condições:** o coordenador municipal está logado e com o *token* de validação armazenado;
- **Pós-condições:** apresentação das demandas de amostras dos laboratórios.
- **Cenário principal:**
 1. O coordenador geral acessa a página demandas de amostras;
 2. O sistema consulta as amostras encaminhadas aos laboratórios e exibe o total de amostras aguardando por exame e número de amostra por laboratório;

A.2.19 Resumo do Trabalho Diário da Equipe

O coordenador municipal pode consultar um resumo do trabalho diário de todas as equipes de uma atividade.

- **Pré-condições:** o coordenador municipal está logado e com o *token* de validação armazenado;

- **Pós-condições:** resumo do relatório do trabalho diário gerado e apresentado.

- **Cenário principal:**

1. O usuário acessa a página de equipes;
2. O sistema exibe a tela com todas as atividades em aberto;
3. O usuário escolhe a atividade e equipe que deseja o resumo;
4. O sistema consulta os dados do dia e apresenta o resumo ao usuário.

A.2.20 Resumo do Trabalho Diário

O agente pode visualizar o resumo do trabalho diário por ele realizado.

- **Pré-condições:** o agente está logado e com o *token* de validação armazenado;
- **Pós-condições:** relatório gerado.
- **Cenário principal:**
 1. O usuário acessa a página de resumo do trabalho diário;
 2. O sistema consulta os dados gerados no dia e apresenta resumo do relatório ao usuário;

A.2.21 Registrar Dados Abertos Na FIWARE

A API faz o registro dos dados coletados durante a vistoria na FIWARE.

- **Pré-condições:** a atividade finalizar trabalho diário foi iniciada;
- **Pós-condições:** os dados da vistoria foram armazenados na FIWARE.
- **Cenário principal:**
 1. O usuário solicita finalizar trabalho diário;
 2. O sistema salva as vistorias na base de dados da ferramenta e na FIWARE;

A.2.22 Imprimir formulário de vistoria

O sistema deve provê a forma de impressão do formulário de vistoria para os agente levarem a campo.

- **Pré-condições:** o agente ou supervisor está logado e com o *token* de validação armazenado;
- **Pós-condições:** arquivos do formulário de vistoria gerado e pronto para impressão.

- **Cenário principal:**

1. O usuário solicita o arquivo do formulário de vistoria;
2. O sistema consulta a metodologia e objetivo da atividade e gera arquivo de acordo com a atividade;
3. Usuário solicita a impressão;
4. O sistema envia requisição de impressão para a impressora e emite mensagem de sucesso.

A.2.23 Visualizar Boletim de Campo

O coordenador municipal pode visualizar qualquer boletim de campo, um supervisor somente visualiza os boletins da sua equipe e um agente somente os seus.

- **Pré-condições:** o coordenador municipal, supervisor ou agente está logado e com o *token* de validação armazenado;
- **Pós-condições:** apresentação do boletim de campo.
- **Cenário principal:**
 1. O usuário solicita o boletim de campo de um determinado dia;
 2. O sistema consulta as vistorias realizadas pelo agente no dia e apresenta o boletm de campo com todas as vistorias;

A.2.24 Visualizar Relatório Semanal

O sistema deve calcular o relatório semanal que é o somatório de todos os relatórios diários em uma semana epidemiológica.

- **Pré-condições:** o coordenador municipal ou supervisor está logado e com o *token* de validação armazenado;
- **Pós-condições:** relatório semanal gerado.
- **Cenário principal:**
 1. O usuário solicita geração do relatório semanal;
 2. O sistema consulta todas as vistorias realizadas durante a semana epidemiológica informada e gera o relatório semanal similar ao do Anexo B;

A.2.25 Enviar Amostra para Laboratório

É de responsabilidade dos supervisores encaminharem as amostras para exames em laboratórios.

- **Pré-condições:** o supervisor está logado e com o *token* de validação armazenado;
- **Pós-condições:** amostra é registrada como demanda do laboratório.
- **Cenário principal:**
 1. O supervisor acessa a página de encaminhamento de amostra;
 2. O sistema exibe a tela de encaminhamento;
 3. O supervisor informa o código na amostra e escolhe o laboratório que ela irá;
 4. O sistema salva as informações e retorna mensagem de sucesso ao supervisor.

A.2.26 Registro do trabalho diário para o agente

Em casos que o agente não possa realizar o registro do trabalho diário, o supervisor deve registrar as vistorias no sistema para o agente.

- **Pré-condições:** o supervisor está logado e com o *token* de validação armazenado;
- **Pós-condições:** os dados das vistorias foram armazenados.
- **Cenário principal:**
 1. O supervisor acessa a página de registro de trabalho diário;
 2. O sistema exibe a tela de registro;
 3. O supervisor informa o código do agente, hora inicial e final do trabalho diário;
 4. O sistema registra as informações e exibe tela de cadastro de vistoria;
 5. O supervisor registra vistoria;
 6. O sistema salva as informações e retorna mensagem de sucesso ao supervisor.

A.2.27 Visualizar Resumo do Trabalho diário da Equipe

O supervisor pode visualizar o resumo do trabalho diário das suas equipes.

- **Pré-condições:** o supervisor está logado e com o *token* de validação armazenado;
- **Pós-condições:** resumo relatório do trabalho diário por equipe gerado.
- **Cenário principal:**

1. O supervisor acessa a página de equipes;
2. O sistema exibe todas as equipes de responsabilidade do supervisor por atividades em aberto;
3. O supervisor escolhe a equipe e solicita o resumo do trabalho diário;
4. O sistema consulta os dados das vistorias realizadas pela equipe e gera o resumo do relatório diário da equipe.

A.2.28 Visualizar resumo do resultado de laboratório de suas amostras

O supervisor pode consultar um resumo dos resultados das amostras que ele encaminhou aos laboratórios

- **Pré-condições:** o supervisor está logado e com o *token* de validação armazenado;
- **Pós-condições:** resumo dos resultados das amostras gerados.
- **Cenário principal:**
 1. O supervisor solicita resumo dos resultados de suas amostras
 2. O sistema consulta as amostras, encaminhadas pelo supervisor, com exames já registrados, gera o resumo dos resultados e apresenta ao supervisor;

A.2.29 Acompanhar demandas por laboratório pelo supervisor

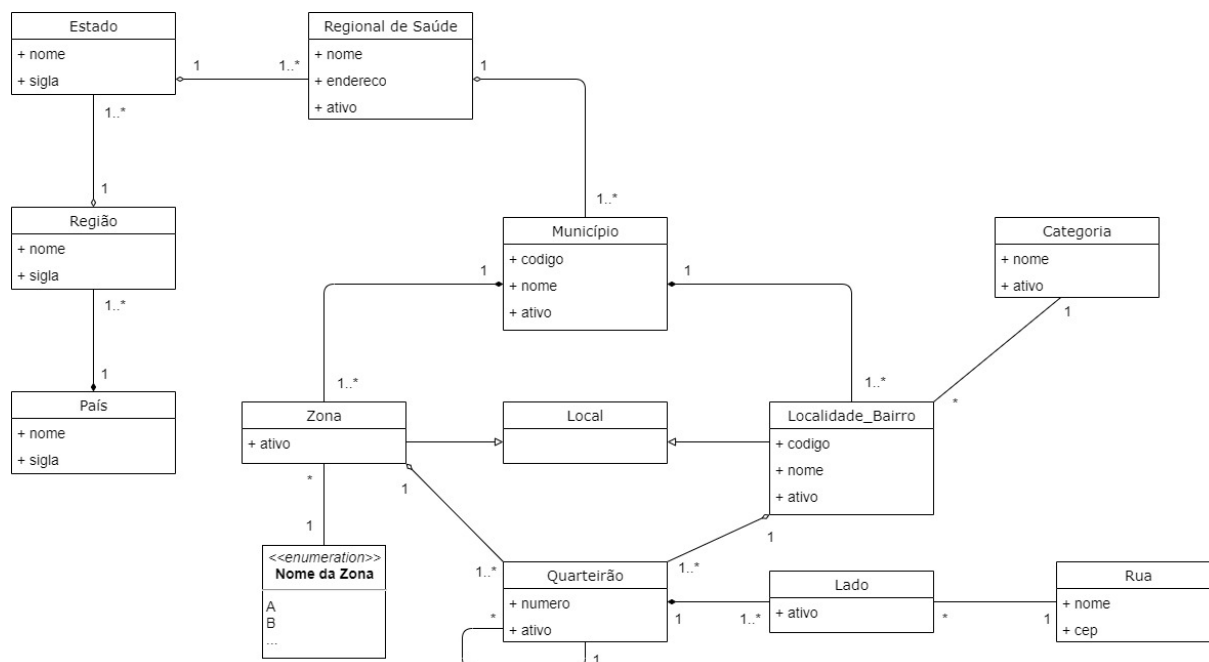
O supervisor pode acompanhar suas demandas aos laboratórios.

- **Pré-condições:** o supervisor está logado e com o *token* de validação armazenado;
- **Pós-condições:** relatório com as demandas por supervisor.
- **Cenário principal:**
 1. O supervisor acessa a página de demandas;
 2. O sistema consulta as demandas do supervisor por laboratório e apresenta o total de amostras em espera e o total por laboratório;

A.3 Diagrama de Classes

Nesta seção iremos descrever as demais visões do diagrama de classes da ferramenta. A [Figura 27](#) descreve a estrutura de local aplicada na ferramenta.

Figura 27 – Visão de Local do Diagrama de Classe da ferramenta AaMT

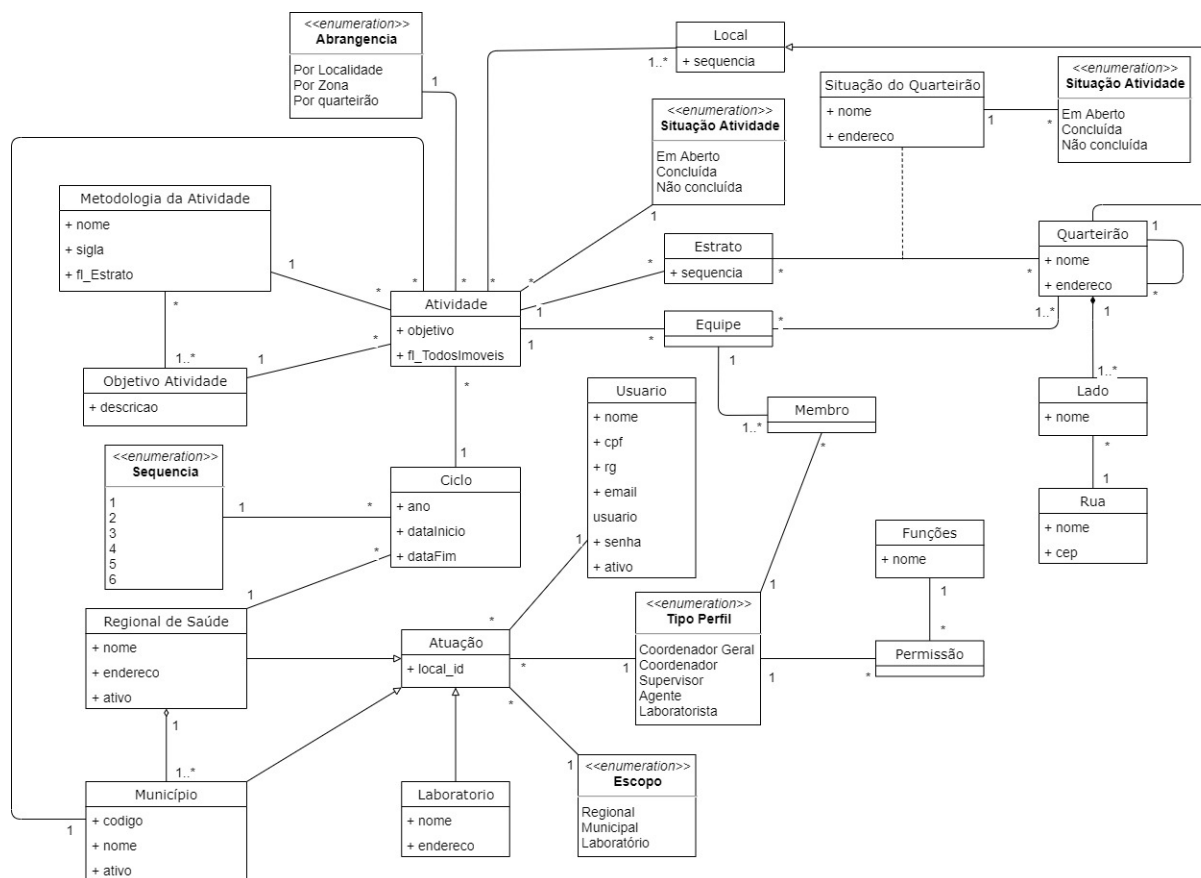


Fonte: os autores

Foi projetado a localidade desde o país até o bairro e as ruas pertencentes a um município. Como mostra a [Figura 27](#), as classes Zona, Localidade/Bairro e Quarteirão são uma generalização de Local, isso é necessário, pois ao cadastrar uma atividade é necessário definir a abrangência da atividade, este atributo refletir em qual generalização da classe Local será aplicada.

Já na visão de atividade, na [Figura 28](#), descreve a estrutura de classes necessárias para solucionar o problema de realizar uma atividade nos municípios. Ao cadastrar uma atividade é necessário informar a metodologia e o objetivo da metodologia que será aplicado na atividade, essas informações é usada para a produção do formulário de vistoria correto durante um trabalho de campo.

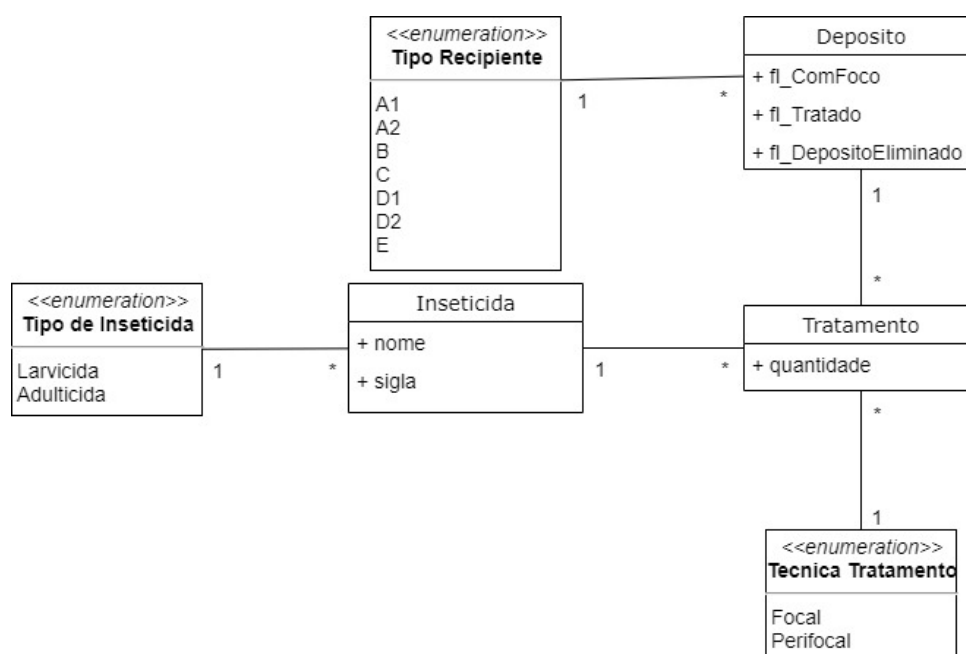
Figura 28 – Visão de atividade do Diagrama de Classes da ferramenta AaMT



Fonte: os autores

Além disto, uma atividade só é considerada concluída quando todos os quarteirões estão com a situação de concluído na classe Situação do Quarteirão. Durante a execução de uma vistoria é realizado a coleta de amostras ou tratamento dos recipientes, isto depende de qual metodologia é aplicada e o seu objetivo. Para isto, foi planejado os componentes de tratamento de recipiente e amostras, descritos na [Figura 29](#) e na [Figura 30](#).

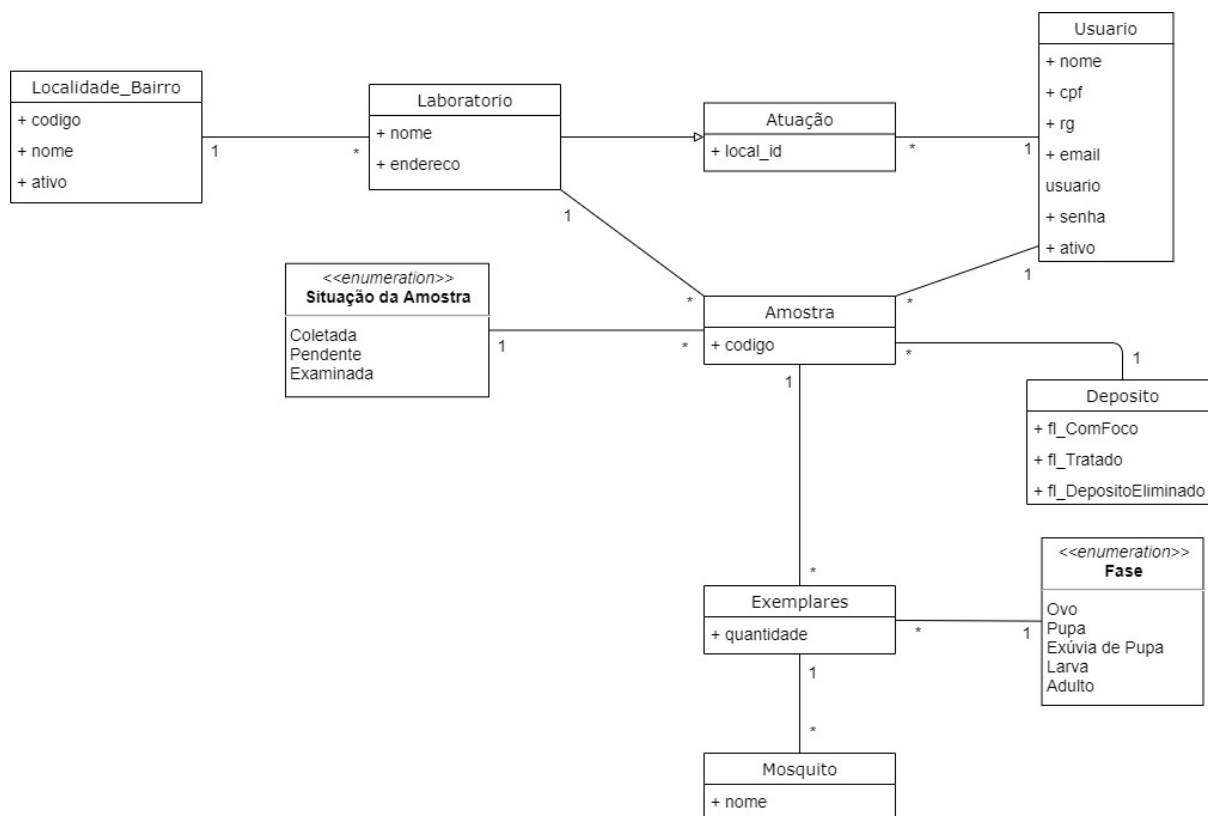
Figura 29 – Visão de tratamento do Diagrama de Classes da ferramenta AaMT



Fonte: os autores

Durante o registro de tratamento, o agente deve informar se o deposito contém foco, se sim, deve ser informado qual o tratamento aplicado, eliminação ou tratamento. A eliminação do deposito consistem em eliminar a água contida no recipiente, já o tratamento é a aplicação de inseticida no recipiente com foco.

Figura 30 – Visão de amostra do Diagrama de Classes da ferramenta AaMT



Fonte: os autores

Uma amostra é coletada de um depósito com o objetivo de realizar exames. As amostras coletadas são enviadas para um laboratório para o registro dos exemplares existente na amostras como: quantidade de ovos, pupas, exúvias de pupa, larvas e adultos.

APÊNDICE B – Exemplo de código, em jQuery, de requisições a API ORION

Este documento visa apresentar códigos desenvolvidos utilizando a linguagem Javascript com o framework jQuery. Os códigos dão exemplo de requisições feitas a API ORION com os padrões exigidos pela NGSI v2 da FIWARE.

Código 1 – Código JavaScript - Consultando entidade por conteúdo dos atributos

```

1  var settings = {
2    "async": true,
3    "crossDomain": true,
4    "url": "http://localhost:1026/v2/entities"
5    + "?type=Restaurant&q=capacity>=50&options=keyValues",
6    "method": "GET",
7    "headers": {
8      "cache-control": "no-cache",
9      "postman-token": "25e32aa7-dc29-ac3e-4e96-df720f03158d"
10   }
11 }
12
13 $.ajax(settings).done(function (response) {
14   console.log(response);
15 });

```

O [Código 1](#) realiza uma requisição do tipo GET, conforme mostra a linha 6, para que seja retornado entidades de contexto de acordo com seus valores de atributos. A especificação deste tipo de requisição é feita na linha 4 onde a variável recebe a URL de destino e parâmetro de especificação NGSI v2 que são descritos na linha 5 a partir do símbolo “?”.

Esse símbolo (“?”) representa, no padrão NGSI v2, que a requisição está passando parâmetros para filtrar as entidades de contexto onde podemos ver que “*type=Restaurant*” implicará na API ORION em retornar somente as entidades que forem do tipo *Restaurant*. O símbolo “&” é utilizado para passagem de mais de um parâmetro, no código 4 por exemplo são passados três valores (descritos na linha 5): *type*, *q* e *options*.

O valor *q* foi usado para filtrar as entidades de acordo com o atributo *capacity* pertencente as entidades do tipo *Restaurant* e retornar somente as entidades que possuírem a capacidade maior que 50.

Entre a linha 13 a 15 do [Código 1](#) refere-se ao uso da função `.ajax` para envio da requisição a API ORION. Após a execução do [Código 1](#) com sucesso o ORION retornará resultados semelhantes ao [Código 2](#).

Código 2 – Código JSON - Resultado por consulta de conteúdo dos atributos

```
1  [  
2    {  
3      "id": "0115206c51f60b48b77e4c937835795c33bb953f",  
4      "type": "Restaurant",  
5      "address": {  
6        "streetAddress": "Cuesta de las Cabras Aldapa 2",  
7        "addressRegion": "Araba",  
8        "addressLocality": "Alegria-Dulantzi",  
9        "postalCode": "01240"  
10     },  
11     "aggregateRating": {  
12       "reviewCount": 2,  
13       "ratingValue": 4  
14     },  
15     "capacity": 80,  
16     "department": "Franchise1",  
17     "description": "Restaurante de estilo sidreria ubicado en  
18       Alegria-Dulantzi. Ademas ...",  
19     "location": "42.8404625, -2.5123277",  
20     "name": "Elizalde",  
21     "occupancyLevels": 0,  
22     "priceRange": 0,  
23     "telephone": "945 400 868"  
24   },  
25   ...  
26 ]
```

O [Código 2](#) é a mensagem de retorno do [Código 1](#) onde o ORION retorna um vetor em JSON contendo entidades descritas na linha 2 a 23.

Anexos

ANEXO A – Glossário do Boletim de Campo e Laboratório do Levantamento Rápido de Índices - LIRAa

Bairro(s): Bairro onde o estrato esta sendo aplicado, este campo também pode ser usado como localidade, comumente usado por fatores locais;

Folha: ;

Logradouro: Rua, Av., Praça, etc;

Município/UF: Município ou estado onde ta sendo aplicado o LIRAa;

Nº de Quarteirões: Quantidade de quarteirões trabalhados no estrato;

Nº de imóveis: Quantidade de imóveis trabalhados no estrato;

Nº do Quarteirão: Quarteirão referente ao imóvel trabalhado;

Nº/Complemento: Número do imóvel ou complemento;

Out: Outros tipos de imóveis que não são **TB**;

TB: Terreno baldio;

Nº de Recipientes com Foco: quantidade de recipientes encontrados com foco na vistoria, separado por tipo de recipiente;

Grupo A: Armazenamento de água para consumo humano;

A1: Caixa d'água ligada à rede (depósitos elevados);

A2: Depósitos ao nível do solo: consumo doméstico (barril, tina, tonel, tambor, depósito de barro, tanque, poço, cisterna, cacimba);

Grupo B: Depósito móveis;

Vasos/frascos com água, pratos, pingadeiras, recip. degelo, bebedouros em geral, peq. fontes ornamentais, mat. dep. construção, objetos religiosos/rituais;

Grupo C: Depósitos fixos;

Tanques/depósitos em obras, borracharias e hortas, calhas e lajes em desníveis, sanitários em desuso, piscinas não tratadas, fontes ornam., floreiras em cemitérios, cacos em muros, toldos, peças arquitet. (caixas de inspeção/passagem);

Grupo D: Passíveis de remoção/proteção;

D1: Pneus e outros materiais rodantes (manchões/câmaras);

D2: Lixo (recipientes plásticos, garrafas, latas) sucatas em pátios, ferros-velhos e recicladoras, entulhos;

Grupo G: Naturais;

Axilas de fohas (bromélias, etc.), buracos em árvores e em rochas, cascas, restos de animais (cascos, carapaças);

Nº das amostras coletadas: Identificação da amostra coletada, a amostra tem que especificar de qual recipiente foi coletada;

Nº de tubitos: Os tubitos são pertencentes a uma amostra, ou seja, uma amostra pode ser dividida em vários tubitos devido a quantidade de amostra;

Número de Tubitos: Campos referente aos tubitos;

Examinados: Número total de tubitos examinados;

Aegypti: Número de tubitos que possui foco do mosquito *Aedes aegypti*;

Albopictus: Número de tubitos que possui foco do mosquito *Albopictus*;

Número de recipientes Positivos: Campos referentes aos mosquitos;

***Aedes aegypti*:** Registro de recipientes positivos com o mosquito *Aedes aegypti* separado por tipo de recipiente;

***Aedes Albopictus*:** Diferente do *Aedes aegypti*, o LIRAA não necessita registrar o número de recipientes positivos por tipo, basta informar o total de recipientes.

Nº de imóveis positivos para *Aedes aegypti*: somatório total de imóveis que foram encontrado foco do vetor *Aegypti*;

Nº de imóveis positivos para *Aedes Albopictus*: somatório total de imóveis que foram mosquito *Aedes Albopictus*;

Nº de terrenos baldios positivos para *Aedes aegypti*: somatório de **TB** com foco do mosquito *Aedes aegypti*;

Nº de terrenos baldios positivos para *Aedes Albopictus*: somatório de **TB** com foco do mosquito *Aedes Albopictus*;

[illegible]

ANEXO B – Glossário do Boletim de Campo do Programa Nacional de Controle da Dengue - PNCD. Serviço Antivetorial

Município: Município onde está sendo aplicado o PNCD;

Cód. Loc: Código da localidade trabalhada no formulário, este código é disponibilizado por software da própria prefeitura;

Nome da Localidade: Nome descritivo da localidade;

Categ. Localidade: Tipo de localidade trabalhada;

Nº e Nome da Zona (desmembramento): especificação da zona trabalhada, a zona pode ser elaborada pelos supervisores dos serviços de combate a endemias;

Tipo: Campo para informar se o trabalho é realizado na sede ou em outras localidades fora da sede;

Zona Concluída: Campo informativo para registrar se todos os imóveis da zona foram inspecionados ou não;

Data da atividade: Registro da data de início da atividade deste boletim;

Ciclo/ano: Assim como o LIRAA o ACE deve informar o ciclo e o ano que está sendo aplicada a atividade PNCD;

Atividade: tipos de atividades aplicadas no PNCD;

1. LI - Levantamento de Índice: atividade para coletar amostras e levantar os índices de infestação da localidade trabalhada;
2. LI+T - Levantamento de Índice + Tratamento: levantamento de Índices e aplicação de larvicida nos recipientes encontrados com foco;
3. PE - Pontos Estratégicos: imóveis considerados estratégicos, como semitérios;
4. T - Tratamento: aplicação de larvicida;
5. DF - Delimitação de Foco: atividade para isolar localidades com foco;
6. PVE - Pesquisa Vetorial Especial: aplicação de pesquisa vetorial para levantamento de dados;

Nº do quarteirão: Número do quarteirão onde o imóvel se localiza;

Sequência - Segundo campo referente ao quarteirão: Em caso de não existir registro do quarteirão (devido a invasões ou outros motivos) põe-se a sequência a partir do último quarteirão identificado. Exemplo: quarteirão 33, sequência 1;

Lado: Para identificar exatamente em qual quarteirão a casa se encontra (em casos de quarteirão não identificado) deve-se informar o lado do quarteirão, norte, sul, leste, oeste, frente, atrás, esquerda ou direita;

Nome do Logradouro: Descrição da rua, avenida, etc;

Nº: número do imóvel;

Sequência - referente ao imóvel: Assim como os quarteirões, caso o imóvel não possua número deve-se informar a sequência do imóvel a partir do último imóvel com número;

Complemento: Complemento do endereço do imóvel;

Tipo do Imóvel: Referente as categorias de imóveis:

R: Residência;

C: Comércio;

TB: Terreno Baldio;

PE: Ponto Estratégico;

O: Outros imóveis;

Hora da visita: Registro do horário que foi inspecionado o imóvel;

Visita: Referente ao atendimento na vistoria;

N: Visita normal;

R: Visita de recuperação, significa que o imóvel constava como fechado ou recusado e o ACE retornou para realizar a vistoria no imóvel;

Pendência: **F** - Fechado / **R** - Recusa;

Nº de Depósitos Inspeccionados: Quantidade de depósitos inspeccionados separados por tipo de recipiente;

Imóveis Inspeccionados (LI): Quantidade de imóveis inspeccionados no mesmo número de imóveis, como por exemplo em vilas, condomínio, prédio, etc;

Nº da amostra:

Qtde tubitos: Total de tubitos coletados na residência;

Depósitos Eliminados: Total de depósitos eliminados na residência;

Imóveis Tratados: Número de imóveis que foi aplicado o tratamento com larvicida;

Focal: Técnica de aplicação de larvicida;

Tipo (L1): Tipo de larvicida focal aplicado, campo variante podendo surgir novos tipos de larvicida;

Qtde. (Carga): Quantidade de larvicida aplicado no imóvel, normalmente calculado em gramas;

Qtde. depósitos tratados: Total de depósitos tratados com larvicida;

Perifocal: Técnica de aplicação de larvicida;

Tipo: Tipo de larvicida perifocal aplicado, campo variante podendo surgir novos tipos de larvicida;

Qtde. Carga: Quantidade de larvicida aplicado no imóvel, normalmente calculado em gramas;

[illegible]



PROGRAMA NACIONAL DE CONTROLE DA DENGUE - PNCD
RESUMO SEMANAL DO SERVIÇO ANTI-VECTORIAL

01	Controle Digitação
----	--------------------

02	Município	03	Código e nome da localidade	04	Zona
----	-----------	----	-----------------------------	----	------

05	Categ. localid.	06	Tipo	07	Ciclo/ano	08	Data início	09	Data final	10	Concluído?	11	Sem. Epidem.
			1-sede 2-outros		/		/ /		/ /		S-sim N-não		/

12	Atividade		
	1-LI - Levantamento de índice	2-LI+T - Levantamento de índice+Tratamento	3-PE-Ponto Estratégico
	4-T - Tratamento	5-DF-Delimitação de Foco	6-PVE-Pesquisa Vetorial Especial

RESUMO DO TRABALHO DE CAMPO															
13	Total quart. concl.	Nº imóveis trabalhados por tipo						Nº imóveis			23	Amostr. Coletadas	Pendência		
		14	15	16	17	18	19	20	21	22			24	25	26
		Residência	Comércio	TB	PE	Outro	Total	Trat. Focal	Trat. Perifocal	Inspecionados		Recusa	Fechados	Recuperados	

TB - terreno baldio

PE - Ponto Estratégico

Nº depósitos inspecionados por tipo										35	Depósito Eliminado				
27	A1	28	A2	29	B	30	C	31	D1			32	D2	33	E

Depósitos Tratados						Adulticida		44	Total de Agentes na semana	45	Total Dias Trabalhados na semana			
Larvicida (1)			Larvicida (2)			42 <th>43 </th>	43							
36	Tipo	37	Qtde. (Gramas)	38	Qt. dep. trat.	39	Tipo	40	Qtde. (Gramas)	41	Qt. dep. trat.		Qtde. (Cargas)	

RESUMO DO LABORATÓRIO																
Nº depósitos com espécimes por tipo																
	46	A1	47	A2	48	B	49	C	50	D1	51	D2	52	E	53	Total
	com Aedes aegypti															
	com Aedes albopictus															

A1 - caixa d'água (elevado) A2 - Outros depósitos de armazenamento de água (baixa) B - Pequenos depósitos móveis C - Depósitos fixos
D1 - Pneus e outros materiais rodantes D2 - Lixo (recipientes plásticos, latas) sucatas, entulhos E - Depósitos naturais

Nº de imóveis com espécimes, por tipo									Nº de exemplares											
	54	Residência	55	Comércio	56	Terreno Baldio	57	Ponto Estratégico	58	Outros	59	Total	60	Larvas	61	Pupas	62	Exúvia de pupa	63	Adultos
	com Aedes aegypti																			
	com Aedes albopictus																			
	outros																			

64 N° e seq. dos quarteirões com Aedes aegypti				65 N° e seq. dos quarteirões com Aedes albopictus				66 N° e seq. dos quarteirões com Aedes aegypti + Aedes albopictus							
/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/

67	Visto do Supervisor	68	Data do visto
----	---------------------	----	---------------